



**UNIVERSIDAD CARLOS III DE MADRID**  
**ESCUELA POLITÉCNICA SUPERIOR**

Ingeniería Técnica en Informática de Gestión

*Proyecto Fin de Carrera*

**Curso en plataforma de e-learning para el aprendizaje  
del sistema CRS Amadeus**

Autor: Antonio García Robles

Dirección: Prof. Dr. D. Miguel Ángel Patricio  
Prof. D. Francisco García Barrado



## **AGRADECIMIENTOS**

*Gracias a Miguel Ángel y a Francisco, por haberme guiado a lo largo de este proyecto. A mi familia por el ánimo que me han dado durante todo este tiempo, sobre todo a Javi, por hacerlo un poco más fácil.*

*A mis amigos por echarme una mano cuando lo he necesitado, especialmente a Jose y Alex, que han estado ahí desde el principio.*

*Gracias también a Ana y Patricia, que se han portado de maravilla siempre. Y muchas gracias a Silvia, por todo.*



# SUMARIO

<b>1. INTRODUCCIÓN.....</b>	<b>6</b>
1.1. JUSTIFICACIÓN .....	8
1.2. MOTIVACIÓN.....	9
1.3. OBJETIVO .....	10
1.4. MEDIOS ENTORNO .....	10
1.4. FUENTES DE INFORMACIÓN .....	11
<b>2. ESTADO DEL ARTE .....</b>	<b>12</b>
2.1. PLATAFORMAS E-LEARNING.....	12
2.1.1 ¿Qué es el e-learning? .....	12
2.1.2 Claves de e-learning .....	17
2.1.3 Plataformas de e-learning .....	32
2.1.4 Plataforma escogida .....	39
2.2. CURSO EN MOODLE.....	40
2.2.1 Elección del sistema.....	40
2.2.2. HTML y XHTML.....	44
2.2.3. GIFT .....	47
2.3. APRENDIZAJE AMADEUS .....	50
2.3.1 ¿Qué es un CRS? .....	50
2.3.2. Aprendizaje actual.....	54
2.3.3. Uso en simulador.....	54
2.4. APLICACIONES WEB.....	55
2.4.1 ¿Qué son? .....	55
2.4.2. Ejemplos de aplicaciones web .....	55
2.4.3. AJAX .....	56
2.4.4. JSP .....	60
2.4.5. PHP.....	62
2.4.6. Comparativa PHP vs JSP .....	66
<b>3. ARQUITECTURA .....</b>	<b>68</b>
3.1. DISEÑO DEL CURSO .....	69
3.1.2. Ejercicios .....	71
3.2. TECNOLOGÍA .....	75
3.2.1 Requisitos.....	75
3.2.2. Apache Tomcat + MySQL.....	76
3.3. MOODLE.....	78
3.3.1. Instalación .....	78
3.3.2. Estructura del curso.....	78
3.3.3. Presentación del curso.....	79
3.3.4. Carga lectiva.....	80
3.3.5. Ejercicios .....	81
3.4. CURSO HTML/ CSS.....	86
3.4.1. Diseño .....	86
3.4.2. Estándares Web .....	87
3.4.3. Usabilidad.....	88
3.5. SIMULADOR.....	88
3.5.1. Interfaz de la aplicación web- lado cliente .....	91
3.5.2. JSP/ JAVA- lado servidor .....	93
3.5.3. Conexión con ejercicios moodle. ....	109
<b>4. EXPERIMENTACIÓN/ PRUEBA CONCEPTO.....</b>	<b>111</b>
4.1. CURSO.....	112
<b>5. CONCLUSIONES .....</b>	<b>115</b>
5.1. CONCLUSIONES .....	115
5.2. LÍNEAS FUTURAS .....	116
<b>APÉNDICES.....</b>	<b>118</b>
<b>BIBLIOGRAFÍA .....</b>	<b>133</b>

## ÍNDICE DE FIGURAS

Ilustración 1. Tabla comparativa de los LMS y LCMS.....	19
Ilustración 2. Funcionamiento básico de JSP .....	60
Ilustración 3. Relación de versiones de Tomcat con especificaciones de Servlet/JSP...	62
Ilustración 4. Base del funcionamiento de la aplicación. ....	68
Ilustración 5. Presentación de un tema del curso. ....	81
Ilustración 6. Ejemplo de cuestionario. ....	82
Ilustración 7. Los primeros temas del curso en Moodle.....	85
Ilustración 8. Imagen del interior de un tema (detalle de códigos de transacción) .....	86
Ilustración 9. Imagen del interior de un tema (detalle de pantallas del terminal) .....	87
Ilustración 10. Ejemplo de código de transacción.....	88
Ilustración 11. Esquema del flujo de datos en el simulador. ....	90
Ilustración 12. Terminal del simulador.....	91
Ilustración 13. Carga de funciones desde JSP (responder.jsp).....	93
Ilustración 14. Flujo de I/O del simulador.....	95
Ilustración 15. Diagrama de paquetes Java .....	96
Ilustración 16. Diagrama de clases Java.....	97
Ilustración 17. Carga de ejercicios. ....	103
Ilustración 18. Diferentes caminos de un ejercicio tipo. ....	104
Ilustración 19. Imagen de un estado de un ejercicio XML.....	105
Ilustración 20. Esquema de flujo para acceder de un estado a otro.....	106
Ilustración 21. Calificaciones en Moodle .....	110
Ilustración 22. Pantalla de <a href="http://www.giaa.inf.uc3m.es/moodle">www.giaa.inf.uc3m.es/moodle</a> .....	111
Ilustración 23. Imagen de un tema de teoría.....	111
Ilustración 24. Los primeros temas del curso. ....	113
Ilustración 25. Preguntas del cuestionario.....	114
Ilustración 26. Ejercicio en el terminal del simulador.....	114



# 1. INTRODUCCIÓN

Con este proyecto se pretende introducir un sistema de aprendizaje online para facilitar el trabajo de enseñanza sobre el sistema Amadeus (un GDS/CRS: sistema de reservas por ordenador).

## 1.1 Justificación

Como proyecto de ayuda a la docencia surge la idea de crear un curso de elearning sobre Amadeus, para así complementar el contenido de las clases de la asignatura Informática Aplicada al Sector Turístico. Con el reciente cambio de planes de estudio se crea el marco perfecto para introducir este curso.

Para motivar el aprendizaje se implantará un sistema de e-learning basado en lecciones y ejercicios tipo test. Todo ello se complementará con un Simulador del entorno Amadeus que requerirá conocimientos aplicados aprendidos en las diferentes lecciones y que formará parte de la calificación final del curso.

El diseño del curso será eminentemente práctico, enfocado a un aprendizaje rápido y directo de cada parte teórica y apoyando la comprensión desde los ejercicios.

Se buscará crear una aplicación de simulación fiel al entorno real. Éste ofrecerá un funcionamiento fiel a la realidad del sistema Amadeus. Tendrá dos características principales: facilidad de uso para el profesor, esto es, una forma sencilla de realizar y/o modificar ejercicios de simulación, y escalabilidad en el diseño, de tal forma que se puedan agregar nuevas funcionalidades de manera similar a librerías, para así complementar en futuras mejoras, si se dieran o necesitasen, la función de simulación. No se pretende lograr una gran exactitud en la representación de la interacción entre el usuario y los ejercicios del simulador, si no más bien ofrecer un marco de aprendizaje que sirva de introducción a futuros estudios o cursos con un contenido más avanzado.

La finalidad del proyecto será ofrecer a los profesores y alumnos una herramienta de aprendizaje sencilla de usar que facilite su labor sin comprometer las necesidades de la enseñanza, dando a los primeros todas las facilidades para adaptar



según sus necesidades o preferencias el desarrollo del curso y a los estudiantes un curso que sirva de introducción a Amadeus con utilidad real y que sienta unas bases suficientemente sólidas para poder mejorar el aprendizaje en el futuro.

## **1.2 Motivación**

### **► Motivación académica**

Las plataformas de e-learning están empezando a ser tenidas en cuenta. Estamos en un mundo en el que las distancias son cada vez más pequeñas gracias a los medios de comunicación, y la educación no puede más que seguir con esa tendencia.

Es cierto que han existido muchas previsiones de que la educación a distancia sustituiría a las clases tradicionales, pero sin embargo esto es algo que no ha ocurrido. Los motivos no están claros, tal vez se trate de la confianza inherente que debe existir entre las personas, la necesidad de relacionarse, o que la improvisación de un buen profesor, sus anécdotas, sus palabras, ayuden a motivar a los alumnos de una forma insustituible por cualquier otro medio.

A pesar de esto el aprendizaje a distancia es una herramienta útil en muchas situaciones, y permite un control directo del tiempo empleado. Son comunes las críticas de los educadores hacia el tiempo que dedican sus alumnos trabajando una asignatura (al escaso o nulo tiempo, me refiero). Con estas herramientas se puede controlar de una forma práctica y sencilla que se hagan los ejercicios requeridos o las lecturas propuestas, por ejemplo.

A pesar de esto pienso que la mejor baza que tiene esta tecnología no es el control que pueda ejercer el profesor sobre los alumnos, si no la motivación extra que encuentran los alumnos al trabajar una asignatura a distancia, pudiendo realizar algunas actividades en cualquier momento, acceder fácilmente a contenido multimedia, y especialmente incluyendo chats y foros donde compartir opiniones sobre la asignatura, con la mayor implicación que ello conlleva.

En definitiva, es una herramienta excelente para complementar con las clases en persona y que puede servir como elemento motivador para el estudiante.

La Universidad Carlos III se ha distinguido siempre por su apoyo a las nuevas tecnologías, contándose entre las universidades que más puestos informáticos destina

a sus estudiantes. Es un acierto rotundo que se decante por esta tecnología y que muestre su apoyo en proyectos y colaboraciones de esta índole, como en la nueva versión de Aula Global 2.

### **1.3 Objetivo**

Se quiere crear un curso sobre una plataforma de e-learning para complementar o suplir el aprendizaje de una asignatura sobre el sistema de reservas Amadeus. Adicionalmente se desea ofrecer una aplicación que haga las veces de simulador del sistema Amadeus para mejorar la comprensión de la asignatura. El simulador debe estar basado en ejercicios fijos, pero a su vez permitir fácilmente una ampliación para añadir, de forma modular, diferentes mejoras a la aplicación.

Para la realización del proyecto se requerirá trabajar conjuntamente sobre distintas plataformas y usar diversos recursos. Las plataformas funcionarán conjuntamente ofreciendo cada una sus capacidades complementarias a las otras, formando en conjunto la aplicación con la funcionalidad deseada.

### **1.4 Medios entorno**

#### **► A QUIÉN VA DIRIGIDO**

- Profesores

Este curso ayudará a los profesores a controlar el desarrollo de los alumnos en un tema complejo, como es el aprendizaje del sistema Amadeus, y permitirá un seguimiento cercano del devenir del curso para ayudar en los momentos necesarios y anticipar posibles problemas e inconvenientes que puedan surgir.

- Estudiantes

De cara al alumno los cursos online facilitan enormemente al autogestión del tiempo y del esfuerzo. En ese sentido ofrecen un beneficio doble, al obligar a organizar un método de estudio sin tener necesariamente una referencia fuera del curso online, además de presentar el contenido del curso en sí.

Los contenidos y la forma de presentarlos serán muy familiares a los alumnos. Nos encontramos en un punto en el que los computadores forman parte del día a día, y cualquiera sabrá desenvolverse en un entorno web sin demasiados problemas.

Por parte del alumno sólo será necesario un ordenador con conexión a internet, la potencia del e-learning radica principalmente en esto, la ausencia de necesidades costosas a nivel de usuario.

Respecto al sistema será necesario un servidor accesible y sencillo. Sin requerimientos altos de ancho de banda o de espacio de almacenamiento. En puntos posteriores se verá qué tecnología se usa finalmente y cuáles son los requisitos que trae esto consigo.

## **1.4 Fuentes de Información**

Para la elaboración de este documento, y por lo tanto del Proyecto de Fin de Carrera, se han utilizado fuentes de información formales e informales, con especial relevancia de estas últimas.

Las fuentes formales han sido principalmente libros divulgativos sobre diferentes tecnologías y artículos especializados sobre temas concretos. Las informales comprenden un amplio abanico de orígenes, como pueden ser blogs, foros y webs dedicadas al mundo de las TIC y del e-learning. La utilidad de estas fuentes viene dada por el entorno cambiante y especializado del aprendizaje online, y por las grandes comunidades que lo soportan, desarrollando contenidos, guías y tutoriales de todo tipo.

En todo caso se ha pretendido siempre obtener la fuente primaria de información, dejando de lado otras posibilidades (artículos sin referencias, contenido directo de Wikipedia<sup>1</sup> etc.). Ante la posible eventualidad de que el contenido online cambiase o desapareciese se informa en cada referencia de la fecha de consulta.

---

<sup>1</sup> La Wikipedia tiene como base el no considerarse una fuente primaria, por lo que no es un origen fiable de información. No obstante suele contener referencias a artículos que sí lo son.

## 2. ESTADO DEL ARTE

Con este proyecto se pretende introducir un sistema de aprendizaje online para facilitar la enseñanza del sistema Amadeus (un GDS/CRS sistema de reservas por ordenador). Este sistema se estudia en la Titulación de Turismo (pendiente de la modificación por los planes de Grado), como Informática Aplicada al Sector Turístico.

Para motivar el aprendizaje se implantará un sistema de e-learning basado en lecciones y ejercicios tipo test. Todo ello se complementará con un Simulador de entorno Amadeus que requerirá conocimientos aplicados aprendidos en las diferentes lecciones y que entrará en la valoración final.

Vamos a comenzar estudiando las diferentes plataformas de e-learning existentes en el mercado, con preferencia por las que son parcial o totalmente de uso gratuito. Luego se presentarán las cualidades de Amadeus, explicando en qué consiste y qué podemos esperar para el simulador. A continuación veremos las características de las tecnologías que se usarán en la creación del contenido del curso, tanto para la teoría como para los ejercicios. Por último se tratarán las necesidades del simulador y la tecnología y técnicas necesarias para su desarrollo.

### 2.1 Plataformas e-learning

#### 2.1.1 ¿Qué es el e-learning?

La informática se encuentra presente en todos los aspectos de la vida actual, y la educación no es una excepción. En este aspecto el e-learning es el último avance informático en la educación a distancia, ofreciendo la oportunidad de crear un entorno y un ambiente enfocado completamente al estudiante, uniendo las posibilidades específicas de contenidos que tiene el e-learning con todas las posibilidades de Internet<sup>2</sup>.

En este sentido el e-learning muestra contenidos interactivos altamente eficaces para animar al estudiante y facilitan el seguimiento de su trabajo a lo largo del curso. El uso de foros, chats y otros elementos colaborativos resultan muy atractivos para el alumno, reforzando el aspecto social del curso de la misma manera que sucede con las redes

---

<sup>2</sup> BONEU, Josep M. Plataformas abiertas de e-learning para el soporte de contenidos educativos abiertos. *Revista de Universidad y Sociedad del Conocimiento*, 2007, vol. 4, nº 1. Disponible en <http://www.uoc.edu/rusc/4/1/dt/esp/boneu.pdf> [Consulta 03-05-09]

sociales actuales.

Respecto a las redes sociales, el e-learning trata de forma similar, aunque en un entorno más cerrado, el uso de los contenidos y de las colaboraciones interactivas. No es casualidad que el surgimiento con fuerza de las redes sociales se produzca al mismo tiempo que la aparición de las plataformas de e-learning y la aceptación masiva de las mismas.

La gran penetración de Internet en la sociedad ha abierto un mundo de posibilidades entre los que se encuentra el e-learning a una escala masiva. Ya no es solo el uso de este sistema en los centros educativos, si no que cualquiera puede montar un curso y ofrecerlo online a todo el que quiera. Esta libertad ha traído una gran aceptación por parte de todos los usuarios y creadores de contenidos, ampliando la oferta de cursos y creando una comunidad numerosa y muy activa.

Las plataformas de e-learning ofrecen software de servidor que se encarga de gestionar los contenidos, los usuarios y todos los aspectos relacionados con el e-learning. Dentro de cada plataforma se ofrecen los cursos que aprovechan las funcionalidades de la misma para mostrar contenidos atractivos de una forma sencilla y accesible.

Al igual que ocurría con las redes sociales y el e-learning, no es de extrañar que en este momento se hable de “la nube”, el “cloud computing”, las “aplicaciones web” y otros términos que hacen referencia a trabajar online sin la necesidad de instalar aplicaciones en local. En este sentido los navegadores web actuales ofrecen una interfaz excelente para este propósito, lo que unido a la facilidad de trabajo con contenidos multimedia que existe actualmente se traduce en que el e-learning es una oferta tremendamente interesante<sup>3</sup>.

Uno de los vacíos actuales del aprendizaje online es la carencia de estándares. Un curso creado en la plataforma Moodle puede no ser compatible con la plataforma Claroline, lo cual es claramente una desventaja puesto que limita las posibilidades del creador de cursos a una plataforma específica, limitando a su vez la competencia a número de usuarios, puesto que a mayor número más posibilidades hay de que los cursos sean mejores, lo que se traduce en mayor publicidad... cuando el escenario

---

<sup>3</sup> NICHOLS, Mark. E-Learning in Context. *E-Primer Series*. Disponible en <http://akoaotearoa.ac.nz/sites/default/files/ng/group-661/n877-1---e-learning-in-context.pdf>. [Consulta 3-05-09]

preferible sería una competencia basada más en las buenas características de cada plataforma.

Existen, no obstante, propuestas de estandarización, lo que ocurre es que no están aceptadas generalmente, además de que cada plataforma las interpreta de una manera u otra. Aún queda mucho camino por recorrer en este sentido en el mundo del e-learning, pero a la vista de los resultados, las expectativas no podrían ser mejores.

### ► E-learning en la actualidad

Habitualmente es un sistema que se compone de tres partes: los alumnos, el servidor con la información teórica y práctica del curso, y finalmente uno o más profesores que apoyen el devenir del curso.

Se utiliza en innumerables contextos, siendo los más habituales la enseñanza universitaria y los cursos de formación académica externos a la universidad, tanto públicos como de acceso interno para la formación dentro de la empresa.

Dentro de la universidad se proponen dos métodos diferentes de uso. Uno de ellos es como complemento a las clases tradicionales presenciales, ofreciendo la teoría en una plataforma que garantice un acceso fiable, con el complemento de poder realizar determinados ejercicios a distancia. El otro método es como sustituto de las clases presenciales, mediante un profesor/administrador del curso y supliendo la asistencia por foros u otros métodos de consulta interna.

Actualmente el aprendizaje a distancia está en claro crecimiento. Son muchas las empresas que ofrecen cursos de todo tipo (principalmente de temas derivados de la informática, como lenguajes de programación, diseño web, etc., pero también de muchas otras clases, como cursos de cocina o de música), gracias a la facilidad de acceso que ofrecen y a las múltiples herramientas para moderar y dirigir el curso, aunque éste incluya a un gran número de alumnos. Las posibilidades están ahí para ser explotadas, y sin embargo la aceptación de los cursos se da tímidamente en asignaturas pesadas de las grandes titulaciones.

Poniendo como ejemplo a la Universidad Carlos III, una de las más comprometidas con las nuevas tecnologías, observamos que se llega a compromisos intermedios en cuanto a las posibilidades de los cursos online. Estableciendo que esta Universidad está a la vanguardia de las nuevas tendencias en educación, especialmente en el plano de las nuevas tecnologías, vemos que se han desarrollado programas y

aplicaciones propios para alojar contenidos, en ocasiones con foros de asignaturas, pero de una manera poco centralizada y en exceso dependiente de los deseos del profesor. Es absolutamente elogiable la actitud de esta Universidad en este aspecto, especialmente a la vista de otros casos en otras Universidades, pero siempre se puede mejorar.

El e-learning divide a los educadores en dos grupos, unos a favor y otros en contra. Está claro que el aprendizaje en línea tiene un gran valor pedagógico, ofreciendo la posibilidad de dividir en partes y subapartados el contenido de una asignatura, pudiendo leerlo o realizar los ejercicios propuestos en cualquier momento desde cualquier lugar. Sin embargo son muchos los educadores que prefieren una clase presencial frente a unas lecciones online.

Si se enfoca una posible raíz del problema estamos enfrentando un constructivismo social que fomenta las relaciones entre alumnos para el estudio de una materia, en este caso mediante la tecnología, frente a un diseño pedagógico establecido por uno o varios educadores que guían a los alumnos en una dirección concreta. El eLearning bebe de estos y otros tantos conceptos para crear las herramientas que faciliten la educación, dejando al educador la libertad de adaptarlos según sus preferencias, pero ofreciendo al mismo tiempo muchas facilidades al alumno.

### ► VENTAJAS E INCONVENIENTES

Puesto que ya hemos hablado del e-learning, se tratará de explicar de manera rápida cuáles son sus ventajas e inconvenientes. Primero las ventajas:

- Facilidad de uso. Simplemente se necesita acceso a Internet, ningún conocimiento adicional.
- Comunicación. El acceso a todo tipo de documentos multimedia como videos o música, añadido a la búsqueda de contenidos relacionados en otras webs u otros entornos favorece enormemente al e-learning.
- Acceso online. La Web nos permite conectarnos a un curso creado en el otro lado del mundo de forma transparente y sin requerir ninguna acción por nuestra parte ni costes adicionales. La globalidad del acceso es una ventaja increíble frente a los medios más tradicionales.
- Bajo coste. Tanto el profesor como el alumno no necesitan hacer una fuerte

inversión para crear o acceder a un curso. Los costes de mantenimiento del servidor suelen ser pequeños, lo que permite a su vez rebajar las tarifas de acceso a los cursos (cuando no son gratuitos), que se traduce en mayores facilidades de acceso al conocimiento.

- Centros de estudios. Actualmente la mayoría de los grandes centros de estudios, como las universidades, tienen en Internet un foco de actividad constante que puede ser cubierto con las plataformas de e-learning. La gestión de contenidos, estudiantes y el seguimiento continuo que ofrecen son muy útiles en este ámbito.
- Interactividad añadida. Los lenguajes del lado del servidor como PHP, JSP o ASP, unidos a Javascript, Java, Flash... dan ilimitadas posibilidades en la creación de contenidos web.
- Expansión. Internet, qué duda cabe, está en continua expansión, es un medio de comunicación universal.
- Flexibilidad. El e-learning ofrece una flexibilidad imposible en el aprendizaje convencional. Los usuarios pueden planificar su tiempo completamente, los profesores ajustar las opciones que ofrezca el curso y mantener un seguimiento puntual o una implicación constante y mayor atención a las acciones de los estudiantes. En el entorno empresarial es igualmente útil, puesto que los cursos de formación no suponen, usando las bases del e-learning, quitar tiempo de tareas importantes o cortar la jornada laboral.
- Aprendizaje. Dependiendo del contenido del temario, es posible mejorar los tiempos de aprendizaje necesarios para completar el curso respecto al método convencional, lo cual es algo a tener en cuenta. Del mismo modo hay cursos que difícilmente podrían impartirse mediante e-learning.
- Comodidad. Realizar un curso online evita desplazamientos, necesidad de configuraciones o programas adicionales (dependiendo del caso).
- Actualización. El contenido de los cursos puede actualizarse de forma inmediata, se puede añadir cualquier contenido y estará disponible al instante para el usuario.
- Personalizado. Los cursos de e-learning pueden ser personalizados, ofreciendo contenidos en función del perfil del usuario y de forma completamente



transparente.

- Seguimiento. El profesor podrá realizar un seguimiento constante de las evoluciones de cada uno de sus estudiantes, prestando atención a los que crea más necesitados de ayuda, conversando con ellos vía chat, foros o mail. Es posible, incluso, contabilizar las conexiones al curso, tiempos y todo tipo de datos relacionados de una forma que no se ha tenido en cuenta hasta ahora.

En cuanto a las desventajas más claras tenemos:

- Desorientación. A pesar del diseño y el énfasis puesto en la usabilidad es posible que el alumno tenga dificultades usando la plataforma. Hay que tener en cuenta que tiene gran cantidad de opciones, aunque la parte más interesante para el alumno sea la principal del curso, hay multitud de enlaces que dan acceso a información adicional, lo que puede resultar complicado de entender al principio.
- Idioma. La comunidad de hispanohablantes es muy numerosa, no obstante el idioma predominante en Internet es el inglés. No es una barrera insalvable dado que el número de cursos creados en español es realmente grande.
- Finalidad de uso. Como todo en la Web, las plataformas de e-learning se pueden usar para fines delictivos o de dudosa utilidad. En un entorno que maneja razonables cantidades de dinero como es la educación, hay que tener especial cuidado.
- Banda ancha. El crecimiento de la Web no ha traído consigo una mejora en las comunicaciones, al menos no directamente. España está a la cola de Europa en cuanto a infraestructuras y precios de acceso a Internet, y si nos ceñimos al mercado de hispanoparlantes la calidad de las conexiones en muchos países iberoamericanos deja mucho que desear. Esto es especialmente limitante respecto a los contenidos multimedia, que son los que más ancho de banda consumen, y es un factor a tener muy en cuenta.

## 2.1.2 Claves de e-learning

### ► EVOLUCIÓN

Los CMS en los últimos años han progresado en tres etapas evolutivas, que han impactado, cada vez de forma más notoria, sobre la velocidad de creación de

contenidos, el coste, la flexibilidad, la personalización del aprendizaje, la calidad en la atención del estudiante y las ventajas competitivas de las organizaciones que han aplicado las soluciones de e-learning.

- a) Primera etapa: los CMS (content management system o course management system) son dentro de las plataformas de e-learning los más básicos y permiten la generación de sitios web dinámicos. El objetivo de estos programas es la creación y gestión de información en línea (textos, imágenes, gráficos, vídeos, sonido, etc.). También se caracterizan por no poseer herramientas elaboradas de colaboración (foros, chats, diarios, etc.) ni apoyo en tiempo real.
- b) Segunda etapa: los LMS (learning management system) aparecen a partir de los CMS y proporcionan un entorno que posibilita la actualización, mantenimiento y ampliación de la web con la colaboración de múltiples usuarios. Están orientados al aprendizaje y la educación, proporcionando herramientas para la gestión de contenidos académicos, permitiendo mejorar las competencias de los usuarios de los cursos y su intercomunicación, en un entorno donde es posible adaptar la formación a los requisitos de la empresa y al propio desarrollo profesional. Disponen de herramientas que permiten la distribución de cursos, recursos, noticias y contenidos relacionados con la formación general.
- c) Tercera etapa: los LCMS (learning content management system) son plataformas que integran las funcionalidades de los CMS y los LMS, que incorporan la gestión de contenidos para personalizar los recursos de cada estudiante y donde las empresas se convierten en su propia entidad editora, con autosuficiencia en la publicación del contenido de una forma sencilla, rápida y eficiente, resolviendo los inconvenientes de las anteriores plataformas. Ofrecen facilidad en la generación de los materiales, flexibilidad, adaptabilidad a los cambios, control del aprendizaje y un mantenimiento actualizado del conocimiento.

Los LCMS añaden técnicas de gestión de conocimiento al modelo de los LMS en ambientes estructurados y diseñados para que las organizaciones puedan implementar mejor sus procesos y prácticas, con el apoyo de cursos, materiales y contenidos en línea. Permiten una creación muy eficiente por parte de sus desarrolladores, expertos colaboradores o instructores que participan en la creación de contenidos.

A modo de síntesis, se resumen en la siguiente tabla las características más relevantes de los sistemas LMS y LCMS proporcionando una comparativa entre estos dos tipos de sistemas de e-learning:

Usos	LMS	LCMS
Usuarios a los que va dirigido	Responsables de los cursos, administradores de formación, profesores o instructores	Diseñadores de contenidos, diseñadores instruccionales, directores de proyectos
Proporciona	Cursos, eventos de capacitación y está dirigido a estudiantes	Contenidos para el aprendizaje, soporte en el cumplimiento y usuarios
Manejo de clases, formación centrada en el profesor	Sí (pero no siempre)	No
Administración	Cursos, eventos de capacitación y estudiantes	Contenidos para el aprendizaje, soporte en el cumplimiento y usuarios
Análisis de competencias-habilidades	Sí	Sí (en algunos casos)
Informe del rendimiento de los participantes en el seguimiento de la formación	Enfoque principal	Enfoque secundario
Colaboración entre usuarios	Sí	Sí
Mantiene una base de datos de los usuarios y sus perfiles	No siempre	No siempre
Agenda de eventos	Sí	No
Herramientas para la creación de contenidos	No	Sí
Organización de contenidos reutilizable	No siempre	Sí
Herramientas para la evaluación integrada para hacer exámenes	Sí (la mayoría de los LMS tienen esta capacidad)	Sí (la gran mayoría tienen esta capacidad)
Herramienta de flujo de trabajo	No	Sí (en algunas ocasiones)
Comparte datos del estudiante con un sistema ERP ( <i>enterprise requirement planning</i> )	Sí	No
Evaluación dinámica y aprendizaje adaptativo	No	Sí
Distribución de contenido, control de navegación e interfaz del estudiante	No	Sí

Ilustración 1. Tabla comparativa de los LMS y LCMS

## ► CARACTERÍSTICAS DE LAS PLATAFORMAS DE E-LEARNING

Hay cuatro características básicas, e imprescindibles, que cualquier plataforma de e-learning debería tener:

1. Interactividad: conseguir que la persona que está usando la plataforma tenga conciencia de que es el protagonista de su formación.
2. Flexibilidad: conjunto de funcionalidades que permiten que el sistema de e-learning tenga una adaptación fácil en la organización donde se quiere implantar. Esta adaptación se puede dividir en los siguientes puntos:
  - Capacidad de adaptación a la estructura de la institución.
  - Capacidad de adaptación a los planes de estudio de la institución donde se quiere implantar el sistema.
  - Capacidad de adaptación a los contenidos y estilos pedagógicos de la organización.
3. Escalabilidad: capacidad de la plataforma de e-learning de funcionar igualmente con un número pequeño o grande de usuarios.
4. Estandarización: hablar de plataformas estándares es hablar de la capacidad de utilizar cursos realizados por terceros; de esta forma, los cursos están disponibles para la organización que los ha creado y para otras que cumplen con el estándar. También se garantiza la durabilidad de los cursos evitando que éstos queden obsoletos y por último se puede realizar el seguimiento del comportamiento de los estudiantes dentro del curso.

Otras características generales observables en las plataformas de e-learning son:

- a) Código abierto: se habla de software «Open Source», cuando éste se distribuye con licencia para poder ver y modificar el código fuente base de la aplicación.
- b) Plataforma gratuita: el uso de la plataforma no supondrá ningún coste por adquisición o licencia de uso. También existe el caso de las plataformas GPL (general public license) Open Source, donde los desarrolladores de estas plataformas ofrecen apoyo en la instalación y otros servicios de manera comercial.
- c) Internacionalización o arquitectura multiidioma: la plataforma debería estar traducida, o se debe poder traducir fácilmente, para que los usuarios se familiaricen fácilmente con ella.
- d) Tecnología empleada: en cuanto a la programación, destacan en este orden PHP, Java, Perl y Python, como lenguajes Open Source, muy indicados para el

desarrollo de webs dinámicas y utilizados de manera masiva en las plataformas GPL.

- e) Amplia comunidad de usuarios y documentación: la plataforma debe contar con el apoyo de comunidades dinámicas de usuarios, con foros de usuarios, desarrolladores, técnicos y expertos.

### ► HERRAMIENTAS DE LAS PLATAFORMAS

Las actuales plataformas de e-learning ofrecen muchas funcionalidades, que pueden ser agrupadas de la siguiente manera:

#### A. Herramientas orientadas al aprendizaje.

- Foros: los foros de discusión son herramientas que permiten el intercambio de mensajes durante el tiempo que dure un curso (o el que estime el formador). Los foros pueden estar organizados cronológicamente, por categorías o temas de conversación (threads) y permitir o no adjuntar archivos (de un determinado tamaño) al mensaje.
- Buscador de foros: son herramientas que facilitan la selección y localización de los mensajes, entre todos los temas de debate que incluyan el patrón de búsqueda indicado.
- e-portafolio: o portafolio digital o electrónico, es una herramienta que permite hacer el seguimiento del aprendizaje de los participantes, teniendo acceso a los trabajos realizados en sus actividades formativas. Los trabajos pueden estar en diferentes formatos tales como imágenes, documentos u hojas de cálculo, entre otros.
- Intercambio de archivos: las utilidades de intercambio de archivos permiten a los usuarios subir archivos desde sus ordenadores y compartir estos archivos con los profesores u otros estudiantes del curso.
- Soporte de múltiples formatos: la plataforma debe ofrecer soporte a múltiples formatos de archivos, como por ejemplo HTML, Word, Excel, Acrobat, entre otros.
- Herramientas de comunicación sincrónica (chat): para el intercambio de mensajes entre los participantes.

- Herramienta de comunicación asíncrona (correo electrónico o mensajería): un correo electrónico puede ser leído o enviado desde un curso. Las herramientas de correo permiten leer y enviar mensajes desde dentro de un curso, o alternativamente habilitan la posibilidad de trabajar con direcciones de correo externas.
- Servicios de presentación multimedia (videoconferencia, vídeo, pizarra electrónica, entre otros): estos servicios se refieren al uso de videoconferencia entre el sistema y el usuario, o a la comunicación entre dos usuarios cualesquiera. Una pizarra electrónica puede ser utilizada por el profesor con sus estudiantes en una clase virtual; éste es un servicio de comunicación sincrónica entre profesores y estudiantes, tal y como puede ser también la compartición de aplicaciones o el chat de voz.
- Blogs / Notas en línea: herramienta que permite a los estudiantes y profesores efectuar anotaciones en un diario. Éste es aplicable en su aspecto más educativo a través de los edublogs.
- Blogs de asignaturas: en las que el profesor va publicando noticias sobre la misma, pidiendo comentarios de sus alumnos a algún texto, propuesta de actividades, calendario, etc.
- Blogs individuales de alumnos: en los que se les pide escribir entradas periódicas, a las que se les realiza un apoyo y seguimiento, no sólo en los aspectos relacionados con la temática o contenidos tratados sino también sobre asuntos relacionados con derechos de autor, normas de estilo, citación de fuentes, etc.
- Blogs grupales de alumnos: en los que, de forma colectiva, a modo de equipo de redacción, tendrán que publicar entradas relacionadas con las temáticas, estilos y procedimientos establecidos.
- Wikis: son herramientas que facilitan la elaboración de documentos en línea de forma colaborativa. Gracias a los wikis el conocimiento ya no se apoya sólo en las fuentes clásicas, sino que es posible encontrar una diversidad amplia de matices, que lo están haciendo más subjetivo. Un ejemplo bien claro de esto es la Wikipedia.

## **B. Herramientas orientadas a la productividad.**

- Anotaciones personales o favoritos: los bookmarks permiten al estudiante volver fácilmente a una página web visitada. Estas anotaciones pueden ser relativas a un curso o no. En cualquier caso, son anotaciones individuales y de uso privado, aun cuando se pueden compartir.
- Calendario y revisión del progreso: utilidades de calendario que permiten al estudiante planificarse en el tiempo, con relación a las actividades de un curso.
- Ayuda en el uso de la plataforma: herramientas de orientación y ayuda a los participantes en el uso del sistema de aprendizaje. Normalmente incluyen tutoriales, manuales de usuario, ayuda en línea o por teléfono o correo electrónico.
- Buscador de cursos: son herramientas que facilitan la selección y localización de los cursos indicando un patrón de búsqueda.
- Mecanismos de sincronización y trabajo fuera de línea: los estudiantes tienen la posibilidad de trabajar desconectados de la plataforma. Previamente, el estudiante se ha descargado el curso, o parte de él, en su ordenador, y trabaja localmente en el curso, de modo que la próxima vez que acceda a la plataforma, se sincronizará o se actualizará el punto en el que se encontraba el estudiante en su estudio, la última vez que se desconectó.
- Control de publicación, páginas caducadas y enlaces rotos: estas herramientas permiten publicar páginas al llegar a una fecha determinada y no dejarlas accesibles una vez finalizado el plazo de publicación; también realizan comprobaciones para localizar y corregir la existencia de enlaces a páginas inexistentes.
- Noticias del lugar: estas herramientas permiten mantener informado de las últimas novedades al usuario de la plataforma.
- Avisos de actualización de páginas, mensajes a foros y envío automático: cada vez que sucede un evento en la plataforma que concierne al usuario, se genera un mensaje automáticamente avisándole de los cambios que se han producido, de esta forma el usuario es informado puntualmente desde la plataforma de los cambios.

- Soporte a la sindicación de contenidos (RSS, 11 News, PodCast, etc.): estas herramientas permiten incorporar a la plataforma contenidos de forma sindicada, que son ofrecidos desde el exterior o desde la propia plataforma, incluso crear contenidos que pueden sindicarse desde otras plataformas. Gracias a los agregadores o lectores de feeds (programas o sitios que permiten leer fuentes RSS), se puede obtener resúmenes de todos los sitios que se desee. Las RSS se reeren normalmente a contenidos textuales, mientras los PodCast se refieren a la descarga de contenidos multimedia (imagen y/o sonido), mediante sindicación. La sindicación en ambos casos se realiza utilizando archivos en formato XML.

### **C. Herramientas para la implicación de los estudiantes.**

- Grupos de trabajo: los grupos de trabajo ofrecen la capacidad de organizar una clase en grupos, de forma que proporciona un espacio para cada uno de ellos, donde el profesor asigna las tareas o proyectos correspondientes.
- Autovaloraciones: los estudiantes a través de estas herramientas pueden practicar o revisar tests en línea, y conocer sus valoraciones. Éstas no son contabilizadas por el profesor.
- Rincón del estudiante (grupos de estudio): espacios donde el estudiante puede hacer grupos de estudio, clubes o equipos de trabajo colaborativos.
- Perfil del estudiante: son espacios donde los estudiantes pueden mostrar su trabajo en un curso, anunciarse, mostrar su fotografía, preferencias, temas de interés o información personal.

### **D. Herramientas de soporte.**

- Autenticación de usuarios: la autenticación es el proceso a través del cual se proporciona acceso a un usuario a su curso, con nombre de usuario y contraseña. La autenticación también se refiere al procedimiento por el cual se crean y mantienen los nombres de usuario y contraseñas.
- Asignación de privilegios en función del rol del usuario: son las utilidades a través de las cuales se asignan privilegios de acceso a un curso, a sus contenidos y herramientas, en función del rol del usuario; por ejemplo: estudiante, profesor, creador del curso, profesor ayudante, administrador, etc.



- Registro de estudiantes: la inscripción de estudiantes a un curso puede efectuarse de diferentes formas. El profesor puede añadir a los estudiantes a su curso o los estudiantes se pueden autoinscribir o borrar, si estas opciones están permitidas, o las inscripciones pueden ser leídas de una base de datos, entre otras formas.
- Auditoría: las herramientas de auditoría permiten consultar todas las acciones realizadas por los participantes de la plataforma, así como obtener estadísticas sobre su utilización. Estas herramientas las utilizan los administradores para verificar el uso que se hace del sistema.

#### **E. Herramientas destinadas a la publicación de cursos y contenidos.**

- Tests y resultados automatizados: permite a los profesores crear, administrar y evaluar los tests realizados. Estos tests se pueden autocorregir, mostrando la solución, comentarios o explicaciones, si así lo quiere el profesor.
- Administración del curso: las herramientas de administración del curso permiten a los profesores tener un control de la progresión de una clase a través del material del curso. También permite a los estudiantes comprobar sus progresos, con los trabajos, tests, pruebas, etc.
- Apoyo al creador de cursos: ayudas y apoyo a los creadores de cursos en la administración de éstos. Estas ayudas pueden venir a través de foros, ayudas en línea, por teléfono, correo electrónico, etc.
- Herramientas de calificación en línea: son herramientas de ayuda a los profesores, para conocer el seguimiento y trabajo del estudiante en el curso.
- Seguimiento del estudiante: estas herramientas proporcionan un análisis adicional sobre el uso que se hace de los materiales del curso.

#### **F. Herramientas para el diseño de planes de estudio.**

- Conformidad con la accesibilidad: significa estar en conformidad con los estándares que permiten que personas con discapacidades puedan acceder a la información en línea.
- Reutilización y compartición de contenidos: hace referencia a que los contenidos

creados por un curso puedan ser compartidos con otro profesor, en otro curso y otro centro. El sistema debe permitir compartir un sistema de archivos y repositorios de contenidos abiertos.

- Plantillas de curso: utilidades para crear la estructura de un curso en línea.
- Administración del currículum: permite proporcionar un currículum personalizado a los estudiantes, basado en los prerrequisitos del programa educativo o actividades, en trabajos previos o resultados de tests.
- Personalización del entorno (look and feel): la personalización del sistema permite cambiar la apariencia gráfica y cómo se ven los cursos, de esta manera se puede dar la imagen de la institución de los mismos.
- Herramientas para el diseño de la educación: herramientas de ayuda a los creadores de cursos para poder crear secuencias de aprendizaje, plantillas o asistentes.
- Conformidad con el diseño de la educación: conformidad con los estándares (IMS, AICC y ADL) para la compartición de materiales de aprendizaje con otras plataformas de e-learning.

#### **G. Sistemas para la gestión del conocimiento en el ámbito educativo.**

Estas herramientas o sistemas hacen diferente énfasis en la realización de su cometido, algunas inciden más en facilitar el trabajo colaborativo para la generación de conocimiento en la comunidad y otras hacen un mayor énfasis en la generación de estructuras de conocimiento. Según la orientación que tienen, pueden clasificarse de la siguiente manera:

- Sistemas integrales de conocimiento: proveen tanto de mecanismos de trabajo colaborativo, como de organización interna de la memoria común del conocimiento. Normalmente representan la estructura del conocimiento de forma jerárquica, en lo que se conoce como árbol de conocimiento; otro mecanismo empleado en la representación del conocimiento es mediante redes jerárquicas de nodos conectados entre sí por relaciones; en cuanto a la función de facilitar la interacción y colaboración de los usuarios, normalmente se realiza por medio de foros o grupos de discusión.

- Los sistemas mediadores de información: se identifican porque su principal objetivo es proveer a sus usuarios de una interfaz para realizar consultas, generalmente mediante la web, sobre un dominio particular, donde las fuentes de conocimiento son heterogéneas y distribuidas, aunque da la apariencia de estar utilizando un sistema centralizado y homogéneo. Para la descripción e indexación de las fuentes de conocimiento y de su contenido, se utilizan lenguajes que permiten relacionar la información, como es el caso del estándar RDF.<sup>12</sup>
- Librerías digitales o repositorios: son aquellos sistemas que son un ensamblaje de las tecnologías de comunicación y almacenamiento digital de información para reproducir, emular y extender el servicio que proveen las librerías convencionales, como son la colección, catalogación, administración y difusión de información bibliográfica. Estos sistemas pueden recolectar información desde fuentes distribuidas de información y le permiten construir al usuario su propia biblioteca digital. En este tipo de sistemas podemos encontrar por ejemplo: Merlot.<sup>13</sup>
- Sistemas basados en ontologías: la orientación de los sistemas de gestión del conocimiento basados en ontologías es diversa, tanto para el campo de los negocios, administración inteligente de noticias, o propósitos más generales, como la definición de modelos conceptuales o investigación, entre otros. En estos sistemas la estructura del dominio es conocida a priori, por lo que soportan sistemas de búsqueda automática del conocimiento y facilitan la toma de decisiones aplicando un motor de inferencia a bases de datos estructuradas ontológicamente. La elección del vocabulario específico de clasificación no suele ser universal y eso presenta problemas de acceso en muchos casos. Además, son normales los solapamientos de los elementos en más de un sitio.
- Sistemas basados en folcsonomías: intentan dar un giro importante a la forma en que se clásica la información. El término folcsonomía (folc+taxo+nomía) significa «clasificación gestionada por el pueblo». De una forma sencilla, la Red se está llenando de sitios donde los usuarios guardan o clasifican, pero de una forma simple, a cada elemento almacenado; el usuario le asigna una o varias palabras clave (tags), que incluso pueden ser compartidas con otros usuarios. Este sistema, que puede resultar anárquico y poco efectivo en principio, está dando resultados no esperados, sobre todo por la cantidad de personas que terminan interviniendo en el procesamiento de la información y el alto grado de

coincidencia que aparece.

Respecto a las plataformas, se han presentado las herramientas y funcionalidades que pueden ofrecer, que son bastante numerosas y, dada la evolución de las TIC e Internet, están cambiando actualmente la concepción del ciclo de vida de los contenidos. Las plataformas de e-learning permiten integrarse con la nueva web 2.0 que presenta los cambios siguientes:

- Cambia la estructura para pasar de una arquitectura cliente-servidor a un servicio web.
- El principal protagonista deja de ser el editor o web-master para pasar a serlo el propio usuario.
- La web deja de ser estática en la presentación de sus contenidos para pasar a ser dinámica.
- La web se transforma de ser sólo un medio de lectura para pasar a ser de lectura y escritura.
- La unidad mínima de contenido era la página; a partir de la web 2.0, pasa a ser el artículo, mensaje, o post.
- Los contenidos están robando el protagonismo a los aspectos tecnológicos o de diseño. El término contenido, desde el punto de vista educativo, es fundamental en los aspectos siguientes:
- Acceder a contenidos: Internet es la mayor fuente de información y conocimiento que nunca ha existido, por tanto, la universalidad de acceso a Internet es primordial desde el ámbito educativo.
- Crear contenidos: la creación o producción verdaderamente efectiva es la de procesos educativos, en los que los alumnos accedan a la información existente, reflexiones e, incluso, lleguen a sus propias conclusiones.
- Recopilar contenidos: es necesaria la recopilación, clasificación y estandarización de los recursos digitales existentes; para ello, existen ya mecanismos eficientes y universalmente aceptados como IMS, SCORM, AICC, entre otros.

- Conectar contenidos: el aprendizaje es un acto o proceso social, y muy poco de lo que aprendemos es estático o absoluto. Por ello hay que encontrar caminos que conecten lo que sabemos con la gran base de datos que es la web y aprender a partir de esas conexiones.

## ► E-LEARNING 2.0

El e-learning como lo conocemos en la actualidad lleva alrededor de 10 años con nosotros. A lo largo de todo ese tiempo ha crecido de ser una idea radical a algo que es ampliamente reconocido como el camino a seguir. Ha sido adoptado activamente por numerosos centros de estudios, universidades y empresas.

Ahora, el e-learning está evolucionando al mismo ritmo que lo hace Internet y la World Wide Web. Si tenemos la web 2.0 podemos hablar de e-learning 2.0.

Pero antes de hablar de hacia a dónde se dirige el mundo del e-learning debemos tratar de explicar dónde estamos ahora.

Cuando pensamos en los contenidos de aprendizaje lo más seguro es que los agrupemos en “objetos de aprendizaje”. Podríamos describirlos como bloques o átomos con pequeños contenidos que pueden organizarse y agruparse. De esta manera se forman cursos que pueden tratarse como libros o manuales.

Hoy en día el e-learning parte de esa misma base para formar sus cursos. Simplemente traslada a un entorno online esas agrupaciones de objetos de aprendizaje. A raíz de esto han surgido los LMS y los LCMS como plataformas idóneas donde ofrecer cursos. Se han aplicado teorías tradicionales de educación a distancia al mundo online, se organizan los contenidos como se haría en un entorno real y se establecen los mismos pasos y requisitos que se tendrían fuera del curso online (profesores, tiempos, etc.).

## USO ACTUAL DEL E-LEARNING

Los sistemas que promueven los procesos de enseñanza-aprendizaje a través de sistemas de e-learning tienen una gran importancia para consolidar la denominada Sociedad del Conocimiento. Dichos medios abren la puerta para la formación básica o avanzada a una importante cantidad de personas, que pueden ver mejorada su cualificación personal o su situación profesional. Estos sistemas tienen un campo enorme de aplicación ya que la formación puede orientarse de forma complementaria

a nivel de educación primaria y secundaria, de forma complementaria o exclusiva a nivel universitario, de postgrado o de formación continua, y de formación especial a medida en las empresas<sup>4</sup>.

No obstante, el campo del e-learning está en sus fases iniciales y le falta un largo camino por recorrer hasta alcanzar su madurez y consolidación. En este camino se presentan interesantes retos y líneas de investigación en el campo pedagógico y en el campo tecnológico, enriqueciéndose cuando se produce una mezcla de los intereses de ambos colectivos por conseguir unos mejores productos, y empobreciéndose cuando los actores de un extremo se empeñan en ignorar a los del otro.

Tampoco se puede ignorar el factor humano, que se convierte en la pieza más importante cuando se quiere acometer una estrategia basada en la e-formación. En un proceso de enseñanza-aprendizaje ni las plataformas tecnológicas, ni los modelos pedagógicos son el fin sino el medio para conseguir el objetivo último del proceso, esto es, aumentar el conocimiento y la formación de las partes implicadas. De forma que se tiene que tener presente que en los extremos de estos medios se encuentran personas que han de creer en los beneficios que pueden aportar estas soluciones. Sin su aceptación, compromiso y entrega será imposible que el e-learning triunfe en una organización.

Durante los últimos 25 años, el uso de e-learning ha aumentado gradualmente. En Septiembre de 2003, más de 8.300 personas de todo el mundo asistieron a la On line Learning Conference en Denver, Colorado, EE.UU. 700 expositores demostraron herramientas de aprendizaje, sistemas de administración de aprendizaje, servicios de consultoría y productos de e-learning que apasionaron hasta a los más experimentados en esta temática.

John Chambers, CEO de Cisco Systems, el líder global en redes de computación, ha descrito al e-learning como "tan grande y tan poderoso que empequeñecerá el e-mail como aplicación de software, tanto en el uso como en el impacto".

Ya que el aprendizaje es una necesidad clave de los seres humanos y ya que el

---

<sup>4</sup> GARCÍA PEÑALVO, Francisco José. Conclusiones de Francisco José García Peñalvo, Estado actual de los sistemas e-learning. *Teoría de la Educación: educación y cultura en la sociedad de la información*, .2005, vol. 6, nº 2. Disponible en : [http://campus.usal.es/~teoriaeducacion/rev\\_numero\\_06\\_2/n6\\_02\\_art\\_garcia\\_penalvo.htm](http://campus.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_garcia_penalvo.htm) [Consulta 3-05-09]

mundo a nuestro alrededor requiere cada vez más aprendizaje a lo largo de períodos más prolongados de nuestras vidas, veremos a Internet como la fuerza más revolucionaria para cubrir esa necesidad<sup>5</sup>.

## EL FUTURO DE E-LEARNING

Se puede intuir hacia dónde evolucionará el e-learning. En primer lugar se adoptarán todo lo posible los elementos de la web 2.0: redes sociales, contenido multimedia, blogs, etc. La posible integración con cualquier elemento es algo indispensable que supondrá un salto hacia adelante en cuanto a posibilidades educativas. Conceptos como web colaborativa, web semántica, cobran forma cuando se trata de e-learning. No es de esperar que las clases online dejen de lado las clases presenciales, es necesario el trato personal más allá del que se puede ofrecer en la web, pero sí será cada vez más común la integración y alternancia entre ambos mundos<sup>6</sup>.

---

<sup>5</sup> VILA, Jordi. E-learning: la revolución ha comenzado. Tomado de eLearning Workshops (Comunidad de eLearning). Disponible en <http://www.elearningworkshops.com/modules.php?name=News&file=article&sid=323> [Consulta 03-05-09]

<sup>6</sup> Entrevista a Martin Dougiamas, *El País*, 4 de diciembre de 2008. Disponible en: [http://www.elpais.com/articulo/portada/Dougiamas/creador/Moodle/Internet/cambia/educa/elpepissupcib/20081204elpcibpor\\_3/Tes](http://www.elpais.com/articulo/portada/Dougiamas/creador/Moodle/Internet/cambia/educa/elpepissupcib/20081204elpcibpor_3/Tes) [Consulta 3-05-09]

### 2.1.3 Plataformas de e-learning

#### ► MOODLE

Moodle es un sistema gestor de cursos (LCMS) que tiene gran aceptación y popularidad entre los educadores de todo el mundo como una herramienta para crear sitios web dinámicos para sus estudiantes<sup>7</sup>.

El objetivo del proyecto Moodle es ofrecer siempre al educador las mejores herramientas para gestionar y promover el aprendizaje. Tiene características que permiten escalar a entornos muy grandes, de cientos o miles de estudiantes, al mismo tiempo que puede usarse con objetivos mucho más reducidos.

Muchas instituciones lo usan como su plataforma principal para alojar cursos completos, mientras otras lo utilizan como añadido a los cursos presenciales. La gran cantidad de módulos diferentes (foros, wikis, etc.) permiten crear comunidades de aprendizaje colaborativo alrededor de una determinada asignatura o materia, al mismo tiempo que ofrecen una forma de dar contenido a los estudiantes y evaluar sus progresos mediante ejercicios.

Ha sido desarrollado desde 1999 hasta la versión actual que es la 1.9, publicada en marzo de 2008. Ha sido traducido a más de 70 idiomas y su comunidad cuenta con más de 400 000 usuarios registrados en su web oficial.

Puesto que se trata de software distribuido bajo los términos de la licencia GPL<sup>8</sup> permite el desarrollo de módulos y modificaciones por parte de la comunidad de usuarios. Dado que la comunidad es tan extensa Moodle tiene una ventaja importante frente al resto de competidores.

---

<sup>7</sup> Moodle.org. Disponible en: <http://moodle.org/about/> [Consulta 4-05-09]

<sup>8</sup> GNU General Public License. Version 2 junio 1991. Moodle.org. Disponible en: <http://docs.moodle.org/es/Licencia> [Consulta 4-05-09]



## ► ATUTOR

ATutor es un LCMS open source basado en web, diseñado con la accesibilidad y adaptabilidad en mente. Los administradores pueden instalar o actualizar ATutor en minutos, crear diseños nuevos para modificar el aspecto, y extender su funcionalidad fácilmente con módulos adicionales. Los educadores pueden recoger, agrupar y distribuir contenido educativo rápidamente, importar contenidos ya creados y gestionar los cursos online. A los estudiantes se les ofrece un entorno educativo adaptativo completo<sup>9</sup>.

La base de ATutor es Open Source<sup>10</sup>, lo que hace que sea una solución efectiva para que tanto grandes como pequeñas organizaciones ofrezcan sus materiales de formación en la web, o presentar cursos online completamente independientes. La ayuda completa está disponible en la documentación, a través de números de soporte técnico o de los foros públicos. Además dispone de traducciones a diversos idiomas.

---

<sup>9</sup> ATutor: Learning Content Management System. Disponible en <http://www.atutor.ca/atutor/> [Consulta 4-05-09]

<sup>10</sup> Licensing. ATutor. Disponible en: <http://www.atutor.ca/services/licensing.php> [Consulta 4-05.09]

## ► DOKEOS

Dokeos es un entorno de e-learning y una aplicación de administración de contenidos de cursos y también una herramienta de colaboración. Es software libre y está bajo la licencia GNU GPL, el desarrollo es internacional y colaborativo. También está certificado por la OSI y puede ser usado como un sistema de gestión de contenido (CMS) para educación y educadores. Esta característica para administrar contenidos incluye distribución de contenidos, calendario, proceso de entrenamiento, chat en texto, audio y video, administración de pruebas y guardado de registros. Hasta el 2007, estaba traducido en 34 idiomas (y varios están completos) y es usado por más de mil organizaciones.

Las principales metas de Dokeos son ser un sistema flexible y de muy fácil uso mediante una interfaz de usuario sumamente amigable. Ser una herramienta de aprendizaje, especialmente recomendada a usuarios que tengan nociones mínimas de computación cuyo objetivo es la preocupación por el contenido<sup>11</sup>.

Dokeos está escrito en PHP y usa bases de datos en MySQL. La versión actual y estable es Dokeos 1.8.

Dokeos.com es también una compañía belga que provee hospedaje soporte y servicios de e-learning, aparte de la distribución de la plataforma Dokeos. La compañía colabora con la comunidad pagándole a varios desarrolladores.

---

<sup>11</sup> DOKEOS. Disponible en: <http://www.dokeos.com/en/deployment/free> [Consulta 4-05-09]

## ► CLAROLINE

Claroline es una plataforma de aprendizaje y trabajo virtual (eLearning y eWorking) de código abierto y software libre (open source) que permite a los formadores construir eficaces cursos online y gestionar las actividades de aprendizaje y colaboración en la web. Traducido a 35 idiomas, Claroline tiene una gran comunidad de desarrolladores y usuarios en todo el mundo<sup>12</sup>.

Publicada bajo una licencia Open Source (De código abierto o software libre), Claroline permite crear y administrar cursos a cientos de organizaciones de 93 países diferentes y la colaboración de espacios online. Cada sitio del curso proporciona una lista de herramientas permitiendo al profesor:

- Escribir la descripción de un curso
- Publicar documentos en cualquier formato (texto, PDF, HTML, vídeo...)
- Administrar foros tanto públicos como privados
- Desarrollar caminos de aprendizaje
- Creación de grupos de estudiantes
- Preparar ejercicios online
- Administrar una agenda con tareas y fechas límite
- Publicar anuncios (también por e-mail)
- Proponer tareas manejadas a través de la red
- Ver la estadísticas de la actividad de los usuarios
- Usar la herramienta wiki para escribir documentos en colaboración

Adaptable a varios contextos de formación, Claroline no es usada únicamente por escuelas y universidades, también en centros de formación, asociaciones y compañías. La plataforma es adaptable y ofrece un entorno de trabajo flexible y ajustable.

Ha sido desarrollada siguiendo las necesidades y experiencias de profesores. Claroline ofrece una interfaz intuitiva y clara de administración de espacios. La gestión o manejo diario de la plataforma no requiere ninguna habilidad técnica especial. La

---

<sup>12</sup> Proyecto Claroline. Disponible en: <http://www.claroline.net/about-claroline.html> [Consulta 4-05-09]

plataforma se instala rápidamente y el uso de cualquier navegador web permite manejar las distintas partes del curso y la admisión de usuarios con fluidez.

## ► ILIAS

ILIAS es un Learning Management System (LMS) Open Source para el desarrollo y la realización de e-learning basado en web. El software fue desarrollado para reducir los costes de utilización de los nuevos medios de comunicación en la educación y la formación, y para garantizar el máximo nivel de influencia del cliente en el desarrollo del software. ILIAS se publica bajo la licencia GPL y de forma gratuita.

Permite la creación eficiente de los cursos y los materiales del curso. Ofrece herramientas y plantillas para el aprendizaje y el proceso de trabajo integrado, incluyendo la navegación y la administración de contenidos y usuarios. Cada usuario en ILIAS tiene su propio escritorio personal, que recoge todos los recursos que son necesarios para el aprendizaje y ayuda a cumplir con las tareas diarias del curso. El escritorio permite la agregación de características personales como noticias, mensajes personales, Recursos de Aprendizaje, notas, Favoritos, feeds (RSS) y otras informaciones. El alumno dispone de estos bloques de información de acuerdo a sus necesidades<sup>13</sup>.

ILIAS tiene múltiples formas de ofrecer contenidos de aprendizaje. Se pueden cargar múltiples tipos de archivo, y SCORM 2004, SCORM 1.2 y AICC son estándares soportados. ILIAS a su vez incluye un entorno de autoría para crear el XML en el que se basan los módulos de aprendizaje, que pueden incluir imágenes, flash, applets y otros archivos multimedia. Es posible crear glosarios y reutilización de las definiciones en otros módulos de aprendizaje

ILIAS está disponible como software de código abierto bajo la Licencia Pública General de GNU (GPL) y puede ser utilizado sin restricciones. Escuelas, universidades, instituciones educativas y cualquier persona interesada podrá utilizar el sistema de forma gratuita y contribuir a su desarrollo posterior. Los usuarios, autores y administradores sólo necesitan un navegador y acceso a Internet. Varios sistemas operativos como Unix, Linux, Mac OS X y Windows se puede utilizar para el servidor de ILIAS con PHP y MySQL. Todo el software adicional de terceros (por ejemplo, MySQL, PHP, Apache) está disponible como software de código abierto también, y se

---

<sup>13</sup> ILIAS information Center. Disponible en :  
[http://www.ilias.de/docu/goto.php?target=lm\\_392&client\\_id=docu](http://www.ilias.de/docu/goto.php?target=lm_392&client_id=docu) [Consulta 4-05-09]

pueden descargar desde su sitio web de forma gratuita.

Usuarios de todo el mundo ya han contribuido al desarrollo de programas coordinados por el equipo de ILIAS de código abierto, en la Universidad de Colonia.

### ► JCLIC

JClic está formado por un conjunto de aplicaciones informáticas que sirven para realizar diversos tipos de actividades educativas: rompecabezas, asociaciones, ejercicios de texto, palabras cruzadas...

Las actividades no se acostumbran a presentar solas, sino empaquetadas en proyectos. Un proyecto está formado por un conjunto de actividades y una o más secuencias, que indican el orden en qué se han de mostrar.

El antecesor de JClic es Clic, una aplicación que desde 1992 ha sido utilizada por educadores y educadoras de diversos países como herramienta de creación de actividades didácticas para sus alumnos.

JClic está desarrollado en la plataforma Java, es un proyecto de código abierto y funciona en diversos entornos y sistemas operativos<sup>14</sup>.

En las páginas de actividades de la zonaClic se ofrecen dos maneras de acceder a los proyectos JClic:

#### Visualizar las actividades en un applet

Un applet es un objeto incrustado en una página web. Los proyectos que se ven de esta manera no quedan almacenados en el disco duro: JClic los descarga, los utiliza y finalmente los borra.

Si el applet JClic no se pone en marcha correctamente hay que comprobar la configuración del sistema Java del ordenador. Se recomienda también echar un vistazo a la página en la que se explica el proceso de carga de los applets.

#### Instalar las actividades en el ordenador

JClic tiene un asistente que permite descargar las actividades y guardarlas en la

---

<sup>14</sup> JClic. ZonaClic, Departamento de Eucación de la Generalitat de Catalunya. Disponible en: <http://clic.xtec.cat/es/jclic/info.htm> [Consulta 4-05-09]

biblioteca de proyectos del ordenador. La biblioteca se crea la primera vez que se pone en marcha JClic, o cuando se intenta hacer la primera instalación de un proyecto.

El proyecto JClic es una evolución del programa [Clic 3.0](#), una herramienta para la creación de aplicaciones didácticas multimedia con más de 10 años de historia. A lo largo de este tiempo han sido muchos los educadores y educadoras que lo han utilizado para crear [actividades interactivas](#) donde se trabajan aspectos procedimentales como diversas áreas del currículum, desde educación infantil hasta secundaria.

Los objetivos perseguidos al iniciar el proyecto serían:

- Hacer posible el uso de aplicaciones educativas multimedia "en línea", directamente desde Internet.
- Mantener la compatibilidad con las aplicaciones Clic 3.0 existentes.
- Hacer posible su uso en diversas plataformas y sistemas operativos, como Windows, Linux, Solaris o Mac OS X.
- Utilizar un formato estándar y abierto para el almacenaje de los datos, con el fin de hacerlas transparentes a otras aplicaciones y facilitar su integración en bases de datos de recursos.
- Ampliar el ámbito de cooperación e intercambio de materiales entre escuelas y educadores de diferentes países y culturas, facilitando la traducción y adaptación tanto del programa como de las actividades creadas.
- Recoger las sugerencias de mejoras y ampliaciones que los usuarios habéis ido enviando.
- Hacer posible que el programa pueda ir ampliándose a partir del trabajo cooperativo entre diversos equipos de programación.
- Crear uno entorno de creación de actividades más potente, sencillo e intuitivo, adaptándolo a las características de los actuales entornos gráficos de usuario.

La herramienta de programación escogida ha sido [Java](#), y el formato para almacenar los datos de las actividades es [XML](#).

## ► PROYECTO SAKAI

El Proyecto Sakai es un Sistema de gestión de cursos (CMS) libre y open source. Contiene un gran número de herramientas software diseñadas para ayudar a los educadores, investigadores y estudiantes a colaborar online en apoyo de su trabajo, sea impartir cursos, investigación o colaboración general en proyectos.

Para los cursos, Sakai ofrece características para apoyar y mejorar el aprendizaje y la enseñanza. Para la colaboración existen herramientas que ayudan a organizar la comunicación y el trabajo colaborativo y en grupo en campus de todo el mundo. Usando un navegador web los usuarios pueden crear un sitio con las herramientas de Sakai que se adapte a sus necesidades, sin necesidad de conocer el lenguaje HTML.

No obstante, la visión del producto alcanza más allá de las aplicaciones de aprendizaje y enseñanza. Muchas instalaciones de Sakai se incluyen para labores de investigación y colaboración. Para ello existen módulos como Sakaibrary, que permite elnazar recursos bibliotecarios a Sakai.

Sakai es open source, está distribuido libremente bajo la licencia Educational Community License. El acceso al código es extremadamente importante para aquellos que quieren modificar su plataforma de campus online o desean desarrollar nuevas herramientas. La calidad de Open Source es importante para toda la comunidad Sakai, puesto que muchos cambios propuestos y creados por la comunidad pueden ser adoptados para el proyecto base<sup>15</sup>.

### 2.1.4 Plataforma escogida

Para la creación del curso se va a escoger la plataforma Moodle. Con una gran cantidad de instalaciones en todo el mundo es una elección idónea por su fiabilidad y sus características. Además la Universidad Carlos III está utilizando Moodle para varios proyectos, entre ellos Aula Global 2, de modo que ofrece un marco inmejorable para el uso de esta plataforma.

---

<sup>15</sup> Proyecto Sakai. Disponible en <http://www.sakaiproject.org/portal/> [Consulta 4-05-09]

## **2.2 Curso en Moodle**

Se va a diseñar un curso en Moodle para los contenidos del aprendizaje de Amadeus. En los apartados siguientes se hablará de los diferentes puntos que se deben tratar en la realización del curso, y en el apartado de "Arquitectura" se explicará cómo se implementan los objetivos aquí propuestos.

### **2.2.1 Elección del sistema**

De entre todos los sistemas de gestión de contenidos que hay disponibles se ha elegido Moodle. Es un sistema implantado ampliamente en entidades de todos los ámbitos, y cuenta con una comunidad de usuarios muy grande y activa además de un probado funcionamiento y estabilidad.

Las razones son varias, algunas están relacionadas con su condición de Software Libre y otras son inherentes a la plataforma. Para empezar, Moodle es adaptable a las necesidades de cada institución o empresa. Tenemos casos donde las modificaciones de aspecto pedidas han sido tan radicales que a primera vista uno no podría decir que esa plataforma es Moodle. También es fácilmente integrable a infraestructuras existentes y tiene una enorme capacidad de crecimiento. Como referencia, hay instalaciones de 500.000 usuarios con más de 3000 cursos simultáneos.

Una de las diferencias más importantes es la disponibilidad de opciones. Moodle tiene una enorme variedad de herramientas para la creación de cursos, y todas están disponibles de manera gratuita. No hay una "Licencia Básica" con opciones recortadas y una "Licencia Plus" que incluya todos los módulos, de hecho no hay licencia: Moodle incluye todo desde el momento en que se instala.

Creo que la más importante de las diferencias de Moodle es el tamaño y variedad de su comunidad. Hay más de 400.000 usuarios registrados en moodle.org, con decenas de foros donde se discute desde estrategias didácticas hasta propuestas de modificaciones. Lo más interesante es que sólo un 30% se reconoce como "técnico", mientras que el resto declara ser docente, investigador o gestor de proyectos educativos. Esto le aporta mucha riqueza al proceso de crecimiento de la plataforma, ya que las personas que deciden hacia dónde va a evolucionar son de todas las partes del mundo y poseen los más variados orígenes formativos. Dentro de esa misma comunidad se ha generado una cantidad enorme de documentación, con la misma dinámica de las plataformas basadas en wikis, como por ejemplo la Wikipedia.



Comercialmente, el hecho de que la plataforma sea abierta le da al cliente la libertad de cambiar de proveedor de servicios en cualquier momento, sin riesgo de perder sus datos o de tener que realizar costosos procesos de migración. Además, Moodle es uno de los proyectos de Software Libre que más ha desarrollado su aspecto comercial manteniendo una red de unos treinta socios en todo el mundo que se ocupan de brindar servicios a nivel de cada país. Estos “partners” están certificados por Moodle y deben seguir estándares preestablecidos para poder mantener dicho respaldo.

A esta plataforma open source se le han dado muchos otros usos aparte del clásico LCMS. Por ejemplo: una escuela que lo utiliza para organizar actividades y gestionar la comunicación a nivel de Dirección y que posteriormente incorporó a los padres; equipos de investigación que lo usan como punto de encuentro para los investigadores por el mundo; gestión del conocimiento dentro del ámbito empresarial; gestión administrativa de trayectos formativos; integrado con herramientas de gestión de RRHH, etc. La flexibilidad de los espacios dentro de Moodle hace que puedan configurarse desde cursos autoasistidos hasta ámbitos de tutoría para acompañar el desarrollo de habilidades en nuevos empleados de una compañía.

#### ► VENTAJAS DE MOODLE<sup>16</sup>

- Moodle es una herramienta que permite gestionar cursos, temas y contenidos de forma sencilla, en los que se pueden incluir gran variedad de actividades y hacer un seguimiento del trabajo de los estudiantes.
- Se basa en la pedagogía social constructivista (colaboración, actividades, reflexión, etc.).
- Dispone de varios temas o plantillas que permiten al administrador del sitio personalizar colores, tipos de letra y otras opciones a su gusto.
- Se encuentra traducido a más de 70 idiomas.
- Permite a los educadores crear Espacios Virtuales de Aprendizaje en los que desarrollar cursos on-line o utilizarlos de apoyo como complemento a la enseñanza tradicional.
- Está desarrollado en su totalidad bajo software libre, lo que la convierte en una alternativa muy interesante frente a otras opciones.

---

<sup>16</sup> MARCELO GUARDIA, Marcelo. Ventajas y desventajas del Moodle aplicado en el aprendizaje cooperativo. Entrada de blog: TIC en Educación Básica- Grupo 4, Dirección de Informática Académica de la Pontificia Universidad Católica del Perú. Disponible en: <http://blog.pucp.edu.pe/item/26621> [Consulta 5-05-09]

- La duda de un estudiante puede ser conocida y aclarada por cualquier usuario, estudiante o profesor.
- Facilita el aprendizaje mediante el intercambio de opiniones entre alumnos y con el profesor, así como permite que los estudiantes establezcan relaciones más cercanas.
- Su diseño modular y sencillo facilita la comprensión del código fuente así como su modificación.
- El sistema ofrece gran cantidad de información sobre cada estudiante individual, lo que permite al profesor trazar nuevas estrategias de enseñanza.
- Estimula y potencia el trabajo independiente y las actividades no presenciales.
- Disminuye las restricciones temporales y espaciales.

#### ► DESVENTAJAS DE MOODLE

- Minimiza el trabajo docente, se rompe el vínculo entre alumno y profesor, imposibilita las interrelaciones presenciales entre docente y maestro.
- Sensación de aislamiento. Es muy importante que la plataforma a utilizar provea de distintas formas de fomentar la comunicación y colaboración.
- Para los docentes es muy difícil realizar un seguimiento a cada estudiante continuamente, y, por otro lado, es muy complicado explicar determinado tema de forma tal que sea comprensible por estudiantes con distintos niveles de aprendizaje.
- La gestión de roles de las categorías de usuarios es insuficiente. Se muestran los mismos contenidos a todos los usuarios.
- La experiencia de la interacción entre los profesores y los estudiantes está centrada en el sitio de la asignatura.

Moodle ofrece unos activos muy importantes que nos inclinan a elegirlo como plataforma para nuestro curso. En primer lugar se utiliza en muchas y variadas entidades, dispone de una comunidad importante que ayuda a solucionar problemas y da consejos de uso, es un sistema fiable y adaptable a múltiples necesidades, ofrece un gran número de opciones de configuración y, en general, proporciona probablemente las mejores herramientas que se pueden encontrar actualmente en cuanto a LMS. Es perfectamente adecuado para lo que proponemos llevar a cabo.

## ► ENTIDADES QUE USAN MOODLE

Son innumerables las instalaciones de Moodle en todo el mundo. Una simple visita a la sección en su página web<sup>17</sup> indica que no es una moda pasajera, y que la inmensa comunidad de usuarios tiene una fe completa en Moodle.

Muchas universidades españolas están implantando esta plataforma de e-learning masivamente, enfocándola a múltiples proyectos pero sobre todo a la gestión del contenido que ofrecen los profesores a los alumnos. La Universidad Carlos III de Madrid, la Oberta de Catalunya, la Universidad de Salamanca y muchas otras ya lo utilizan como extensión de un Campus Virtual.

Al mismo tiempo las empresas están adoptando también esta plataforma como punto de partida para sus cursos de formación. Entre ellas se encuentran Cisco, Intel o Caixa Sabadell.

En vista de la aceptación que está teniendo el aprendizaje online sólo se puede augurar un futuro prometedor<sup>18</sup>.

---

<sup>17</sup> Registered Moodle sites. Moodle.org. Disponible en: <http://moodle.org/sites/> [Consulta 4-05-09]

<sup>18</sup> MOLIST, Mercè. Moodle llena la geografía educativa española de campus virtuales. *El País*, 4 de diciembre de 2008. Disponible en: [http://www.elpais.com/articulo/portada/Moodle/llena/geografia/educativa/espanola/campus/virtuales/elpeisupcib/20081204elpcibpor\\_1/Tes/](http://www.elpais.com/articulo/portada/Moodle/llena/geografia/educativa/espanola/campus/virtuales/elpeisupcib/20081204elpcibpor_1/Tes/) [Consulta 5-05-09]

### 2.2.2. HTML y XHTML

HTML 4 es un modelo conforme a SGML (Standard Generalized Markup Language) que es acorde con el estándar internacional ISO 8879, y es aceptado ampliamente con el lenguaje estándar de publicación de contenidos en la World Wide Web<sup>19</sup>.

SGML es un lenguaje para describir lenguajes de marcado, específicamente los utilizados en el intercambio de documentos electrónicos, gestión y publicación de documentos. HTML es un ejemplo de lenguaje definido en SGML.

HTML fue concebido originalmente como un lenguaje para intercambiar documentos científicos y técnicos, fácil de usar para los menos especialistas de documentación. Ha resuelto el problema de complejidad de SGML mediante el uso de etiquetas que definen la parte estructural y semántica del documento de una forma realmente simple. Adicionalmente se añadió el soporte para el hipertexto. Las capacidades multimedia vinieron después.

En un sorprendentemente corto espacio de tiempo, HTML fue extremadamente popular, y rápidamente sobrepasó su propósito original. Desde la adopción de HTML como lenguaje común en Internet ha habido un buen número de elementos añadidos para usar en HTML y adaptarlo a nuevas funcionalidades y entornos.

XHTML es una familia de tipos de documentos que reproduce y amplía HTML4. Los documentos de la familia de XHTML están basados en XML, y usados para funcionar tanto en navegadores como en aplicaciones que utilicen XML.

XHTML 1.0 es el primer tipo de documento de la familia XHTML. Es una reformulación de HTML4 como tipos de documentos que usan XML. Pretende ser usado como un lenguaje de contenidos que adopta las reglas de XML y permite respetar las especificaciones de HTML4. Los beneficios que tienen los desarrolladores si usan XHTML son:

- Los documentos XHTML son conformes con el estándar de XML. Pueden ser editados y validados por herramientas de XML.

---

<sup>19</sup> PEMBERTON, Steven [et al.]. XHTML™ 1.0. The Extensible HyperText Markup Language (2ª ed.): a Reformulation of HTML 4 in XML 1.0. *W3C recommendation*, 26 enero 2000. Disponible en: <http://www.w3.org/TR/xhtml1/> [Consulta 7-05-09]

- Los documentos XHTML pueden escribirse para funcionar igual o mejor que con HTML4, al igual que para agentes de usuario que acepten XHTML 1.0.
- XHTML puede utilizar aplicaciones (scripts y applets, por ejemplo) que se apoyan en el DOM (Document Object Model) de HTML o en el DOM de XML.
- Según vaya evolucionando la familia de XHTML, los documentos conformes a XHTML 1.0 funcionarán y se manejarán de forma más sencilla en los diferentes entornos para XHTML.

La familia de especificaciones XHTML es el siguiente paso en la evolución de Internet. El mundo alrededor de XML está lleno de beneficios, al mismo tiempo que ofrece una base sólida para compatibilizar los contenidos antiguos con los que están por venir. Actualmente se está trabajando con la especificación XHTML1.1 y comenzando a definir el futuro en torno a HTML5<sup>20</sup>, que se definirá en los próximos meses.

#### ► LECCIONES EN HTML

Las páginas en html son la base de la web actual. Basando el modelo de aprendizaje en un entorno web parece adecuado ofrecer las lecciones en html. El desarrollar las lecciones en formato web nos posibilita controlar exactamente el formato además del contenido, adecuar éste a cualquier situación y reducir los requisitos técnicos para acceder al curso.

Además la penetración de Internet en la sociedad ayuda que los usuarios estén familiarizados con el entorno y los estándares habituales de estas páginas, por lo que resulta aún más interesante si cabe la presentación de las lecciones en HTML.

#### ► DISEÑO EDUCATIVO

La Organización Internacional para la Estandarización (ISO<sup>21</sup>) dispone de dos definiciones de usabilidad:

---

<sup>20</sup> HTML5 pretende ser una innovación en contenidos multimedia en la web, pero la dificultad de adoptar un estándar de vídeo y audio común a la industria puede reducirlo a cambios menores, y no suponer el impacto esperado. El debate surge entre la posibilidad de usar estándares abiertos como OGG frente a otros cerrados como H.264. En la situación actual de frentes abiertos por el control de los estándares de la alta definición la solución no parece fácil.

<sup>21</sup> ISO (International Organization for Standardization). Disponible en: <http://www.iso.org/iso/home.htm> [Consulta 4-05-09]

- a) ISO/IEC 9126: "La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso". Esta definición hace énfasis en los atributos internos y externos del producto, los cuales contribuyen a su usabilidad, funcionalidad y eficiencia. La usabilidad depende no sólo del producto sino también del usuario. Por ello un producto no es en ningún caso intrínsecamente usable, sólo tendrá la capacidad de ser usado en un contexto particular y por usuarios particulares. La usabilidad no puede ser valorada estudiando un producto de manera aislada (Bevan, 1994).
- b) ISO/IEC 9241: "Usabilidad es la efectividad, eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso específico". Es una definición centrada en el concepto de calidad en el uso, es decir, se refiere a cómo el usuario realiza tareas específicas en escenarios específicos con efectividad.

En general usabilidad se define como facilidad de uso, término que engloba el diseño, colores, formas, colocación de enlaces y menús, cantidad de subpáginas, facilidad de comprensión del entorno... en definitiva todo lo que conlleva una mayor facilidad de uso. A raíz de esto han surgido ideas y reglas "oficiosas" que pretenden simplificar los términos en la búsqueda de una mejor usabilidad (por ejemplo, la idea de necesitar un máximo de tres clics para acceder al contenido buscado en el sitio web). Puesto que se trata de un concepto abstracto y en gran parte subjetivo las interfaces de usuario tienen mucho margen para la innovación.

### 2.2.3. GIFT

El formato GIFT permite crear diferentes tipos de preguntas (elección múltiple, verdadero-falso, respuesta corta, preguntas numéricas...) de una forma sencilla para que puedan ser importadas.

Cuando se crean grandes grupos de preguntas, el formato GIFT ofrece una forma rápida de cargarlas en categorías para crear ejercicios o en lecciones. Muchas veces es más fácil manejar las categorías teniéndolas en un archivo GIFT<sup>22</sup>.

#### ► INSTRUCCIONES GENERALES

Se deja una línea en blanco entre preguntas. En un formulario simple se pone primero la pregunta y después las respuestas entre corchetes, indicando la(s) correcta(s) y la(s) falsa(s). Se pueden poner comentarios comenzando la línea con dobles barras.

Los archivos GIFT deben estar codificados en UTF8 correctamente. Vamos a ver algunos ejemplos:

```
//Comentario
::Título de pregunta
:: Pregunta {
=Respuesta correcta
~Respuesta incorrecta 1
#Comentario a la respuesta incorrecta 1
~Respuesta incorrecta 2
#Comentario a la respuesta incorrecta 2
~Respuesta incorrecta 3
#Comentario a la respuesta incorrecta 3
~Respuesta incorrecta 4
#Comentario a la respuesta incorrecta 4
}
```

El formato más escueto posible para una pregunta de respuesta múltiple es:

```
Pregunta{= Correcta ~Incorrecta1 ~Incorrecta2 ~Incorrecta3 ~Incorrecta4 }
```

#### Elección múltiple

##### Forma corta:

```
Who's buried in Grant's tomb?{=Grant ~no one ~Napoleon ~Churchill ~Mother Teresa }
```

##### Forma larga:

---

<sup>22</sup> GIFT format. Moodle.org. Disponible en: <http://docs.moodle.org/en/GIFT> [Consulta en 8-05-09]

```
// question: 1 name: Grants tomb
::Grants tomb::Who is buried in Grant's tomb in New York City? {
=Grant
~No one
#Was true for 12 years, but Grant's remains were buried in the tomb in 1897
~Napoleon
#He was buried in France
~Churchill
#He was buried in England
~Mother Teresa
#She was buried in India
}
```

### Verdadero-Falso

```
// question: 0 name: TrueStatement
::TrueStatement about Grant::Grant was buried in a tomb in New York City.{T}
```

### Respuesta corta

Dos ejemplos usando las formas posibles de respuesta corta.

```
Who's buried in Grant's tomb?{=Grant =Ulysses S. Grant =Ulysses Grant}
```

```
Two plus two equals {=four =4}
```

### Unir

Para unir se usa el signo igual antes de un elemento y -> antes de la pareja correcta.

```
Match the following countries with their corresponding capitals. {
=Canada -> Ottawa
=Italy -> Rome
=Japan -> Tokyo
=India -> New Delhi
}
```

### Rellenar el hueco

```
Grant {~is not buried =is buried ~might be buried} in Grant's tomb.
```

### Pregunta numérica

Esta es una pregunta numérica simple. Acepta respuestas con una diferencia de 5 años..

```
When was Ulysses S. Grant born?{#1822:5}
Here is a more complex numerical format with a ranged and partial credit given for 1 answer.
```



```
//this comment will be ignored in the import process
::Numerical example::
When was Ulysses S. Grant born? {
=1822:0 #Correct! you will get full credit for this answer
=%50%1822:2 #He was born in 1822.
#You get 50% credit for being close.
}
```

### ► USO DE PREGUNTAS EN FORMATO GIFT

Como se ha visto el formato GIFT ofrece una flexibilidad enorme para trabajar con tandas de preguntas directas en formatos muy variados. Es un complemento ideal para cualquier curso.

Nosotros vamos a aprovecharlo como complemento de los ejercicios propuestos para el simulador. Más adelante veremos cómo se trabajará con las calificaciones del curso, pero pretendemos que los cuestionarios sean una parte importante de la evaluación al alumno.

A diferencia de un ejercicio en otro formato, un cuestionario es simple y directo, obliga al alumno a aproximarse de una forma diferente a como debería hacerlo si fuese un ejercicio más elaborado, y junto con el simulador refuerza el aprendizaje de cada lección.

## **2.3 *Aprendizaje Amadeus***

### **2.3.1 ¿Qué es un CRS?**

#### **► HISTORIA**

Amadeus designa tanto a la empresa Amadeus como al GDS (Global Distribution System) también nombrado como CRS (Computer Reservation System), un sistema de reservas por ordenador. Es un sistema que sirve para coordinar las reservas de las diferentes aerolíneas asociadas al sistema, establecer trayectos, control de aparatos y, principalmente, control de pasajeros.

Un Sistema de Distribución Global (GDS) es un sistema de almacenamiento y consulta de información relacionada con los viajes y el turismo. Se suelen llamar también sistemas de reservas por ordenador (CRS), pero su carácter global y no orientado exclusivamente a un sector le otorga la característica "global". Actualmente los GDS operan con líneas aéreas, hoteles, alquiler de coches, transporte marítimo, etc.

Ahora mismo el foco de mayor interés de los GDS es internet y las nuevas tecnologías, para permitir realizar reservas de una manera más cómoda y sin intermediarios, pero de momento es una opción a largo plazo que no sustituirá a los métodos tradicionales de reserva por los motivos que explicaremos más adelante.

En la actualidad existen cuatro GDS principales: Amadeus, Sabre, Galileo y Worldspan. Los dos primeros son los más importantes, tienen un mayor número de aerolíneas asociadas, y por lo tanto controlan una mayor parte del transporte mundial. Existe una lucha oculta entre estas grandes compañías, puesto que a mayor cuota de mercado mayores beneficios, especialmente entre Amadeus y Sabre. La primera tiene las aerolíneas de la zona europea y países del mediterráneo, la segunda Norteamérica y parte de Asia.

En este curso trataremos el GDS Amadeus y su modo de terminal, aprenderemos a realizar una reserva partiendo de cero: desde la creación de registros de pasajeros (PNR) y visualización de la disponibilidad de un vuelo determinado hasta la emisión del billete.

La empresa Amadeus fue fundada en 1987. Su principal fuente de negocios es su sistema de reservas de viajes, también conocido por el mismo nombre de la empresa:

Amadeus.

Amadeus fue creada por una alianza entre Air France, Lufthansa, Iberia L.A.E., y Scandinavian Airlines System. Se especializa en soluciones informáticas para crear reservas de aerolíneas, trenes, cruceros, alquiler de coches, hoteles y viajes.

El Sistema de Distribución Global (GDS) de Amadeus tiene la red de distribución internacional más extensa del mundo y se hacen más reservas a través de su sistema que a través de cualquier otro sistema de reservas. Se usa por 67,000 agencias de viajes y 10,000 oficinas de ventas de aerolíneas en todo el mundo. Con Amadeus se pueden hacer reservas con 490 aerolíneas, lo que representa más del 95% de los asientos de las aerolíneas del mundo, 70.000 hoteles, 45 compañías de alquiler de coche, además de otros proveedores de viajes (ferris, trenes, cruceros, compañías de seguros y tour-operadores).

Amadeus tiene suscriptores en más de 217 países en todo el mundo y ofrecen soluciones locales para el marketing, la atención y ayuda al cliente a través de una red de más de 70 Compañías Nacionales de Marketing (CNM). Emplea a un equipo de más de 6,500 personas y 95 nacionalidades en todo el mundo, cubriendo las principales ciudades y CNMs consolidadas.

Las oficinas centrales de los Estados Unidos se encuentran en Miami. Las oficinas centrales están en Madrid, España. Otras oficinas importantes se encuentran en Sophia Antípolis, Francia y Erding, Alemania. Cuentan con oficinas de servicios informáticos en Londres, Reino Unido y Sydney, Australia.

Amadeus España: es propiedad al cien por cien de Amadeus IT Group S.A., compañía multinacional con sede central en Madrid, principal proveedor de tecnología y distribución para la industria mundial del viaje y el turismo.

El funcionamiento del sistema Amadeus es el siguiente: un usuario accede a un terminal del sistema, a partir de ese momento todas sus acciones son registradas, y todas sus acciones en el sistema se llevan a cabo desde su terminal al servidor de Amadeus, el cual se encarga de establecer la conexión con los servidores de las aerolíneas. En resumen: el usuario se comunica con el sistema Amadeus, que es quien se encarga de comunicarse con las aerolíneas. La información sobre la disponibilidad de plazas en los vuelos siempre la tienen las aerolíneas, el sistema

Amadeus, dependiendo del acuerdo que tenga con la aerolínea, tendrá un tipo acceso u otro a las reservas que condiciona el modo en el que se hace la reserva.

El acceso a la información de las reservas no siempre es el mismo. En ocasiones el agente puede conectarse con la propia aerolínea para realizar inmediatamente la reserva sin necesidad de confirmación adicional, dependiendo del nivel de Acceso Amadeus que ofrezca la aerolínea para cada vuelo. El funcionamiento de este sistema lo veremos en temas posteriores.

### ► APRENDIZAJE HABITUAL DE AMADEUS

El terminal de Amadeus funciona mediante códigos transaccionales, los cuales al ser introducidos devuelven en pantalla la información. Existen sistemas gráficos (estilo Windows) los cuales son tal vez más simples de dominar, pero al mismo tiempo son menos potentes, además es más sencillo una vez conocido el modo terminal aprender el modo gráfico que no al revés. En este curso vamos a trabajar con el sistema de terminal dejando aparte el modo gráfico.

Con la aparición de nuevas tecnologías en los últimos años la innovación ha aparecido en el mundo de los GDS, sin embargo el modo terminal sigue siendo más utilizado que el gráfico, especialmente en determinadas áreas. Conociendo los comandos se accede mucho más rápido a la información deseada, y hay determinados puestos en los que la velocidad es importante. Además es un sistema mucho más fiable que los sistemas gráficos, lleva bastante tiempo funcionando por lo que asegura la mayor estabilidad y menor número de fallos.

### ► ENFOQUE DESDE EL PROYECTO

En este proyecto vamos a trabajar en tres niveles diferentes de aprendizaje:

El primero de ellos es un nivel teórico. Está compuesto por las lecciones no presenciales accesibles desde la plataforma de e-learning y por las posibles clases presenciales que impartan los profesores de la asignatura. Será complementado con un foro abierto a todos los estudiantes y profesores para consultas y/o añadidos al temario, así como chats u otras herramientas si el profesor lo cree necesario. Se pretende otorgar toda la libertad posible a los educadores.

El segundo es un nivel de ejercicios tipo test. Se pretende con esto lograr una asociación inmediata entre las diferentes nociones y términos vistos en cada lección. Las cuestiones pueden plantearse de cualquier tipo, siendo preferibles las que tengan una base teórica mayor, puesto que la práctica se deja para el siguiente nivel. Sería idóneo establecer límites de tiempo cortos y controlados para realizar los tests, pero esto se verá en próximos apartados. En cualquier caso el profesor tendrá total libertad para modificarlo según las necesidades del curso o sus preferencias educativas.

El tercer nivel es el de ejercicios de simulación. Con una aplicación que funcione como simulador se establecerán sesiones para realizar tareas específicas en un entorno similar al real de un terminal del sistema Amadeus. La aplicación permitirá la creación de ejercicios específicos así como la implementación futura de módulos especializados para realizar mejoras en la simulación.

#### ► UTILIDAD PARA EL ESTUDIANTE

Los entornos educativos online ofrecen un estímulo adicional al estudio, fomentan la cooperación entre alumnos y facilitan el seguimiento del profesor. Es una manera muy atractiva de presentar un curso, y con los elementos adecuados facilita enormemente el aprendizaje.

En el caso que nos ocupa es imprescindible ofrecer un entorno digital para tratar con la mayor fidelidad posible el acceso a Amadeus. Las plataformas de e-learning ofrecen todo lo necesario para garantizar un curso adecuado al tema y para integrar el entorno Amadeus en el aprendizaje.

Además de todas las ventajas educativas el estudiante verá ventajas inmediatas, como son la consulta de notas online, el uso de foros de debate sobre temas de la asignatura o la disponibilidad de un mayor número de recursos teóricos y prácticos en diversos formatos.

El mayor problema que puede existir es la necesidad de adaptación al sistema de e-learning, pero con la penetración que tiene la tecnología en la sociedad actual y especialmente en el ámbito universitario no debería suponer un problema más allá de casos puntuales.

### **2.3.2. Aprendizaje actual**

En la actualidad la forma de aprender a utilizar el sistema Amadeus es a través de academias con mayor o menor dependencia de la compañía y con unos costes relativamente elevados.

Existen varios modelos de aprendizaje, muchos de ellos basados en interfaces visuales, otros basados en la interfaz clásica de texto, pero todos funcionan sobre el mismo sistema. Los cursos se basan en los diferentes aspectos del sistema, tales como las reservas de vuelos, hoteles, etc. Y en cada aspecto profundizan por fases, por ejemplo en las reservas de vuelos primero la búsqueda de disponibilidad, la reserva de asientos, la emisión del billete... La estructura típica de los programas de cursos de Amadeus es fija puesto que, a pesar de que el sistema permite un gran número de opciones, los procedimientos habituales son muy mecánicos y rara vez se salen de unos pocos casos específicos.

Al ser software propietario no existen referencias libres ni cursos gratuitos. En la universidad permiten realizar este curso aunque no ofrecen ningún apoyo específico.

### **2.3.3. Uso en simulador**

Con el trabajo en un entorno que simule el sistema Amadeus se pretende establecer un factor que facilite en un futuro el uso del sistema real, aprovechando la familiaridad que se haya tenido con el simulador, además de, por supuesto, los ejercicios propuestos dentro del propio simulador.

La propuesta del simulador no es nueva, en los entornos de aprendizaje ofertados por amadeus y varias academias se usa software que simula el sistema Amadeus para lograr unos objetivos similares a los que se proponen aquí. La diferencia es que en este proyecto se pretende partir de una pequeña base que en el futuro pueda seguir ampliándose con funcionalidades añadidas, dando libertad completa a los educadores para que añadan ejercicios o funciones extras según lo crean necesario.

La ventaja que ofrece el simulador es que nos da la posibilidad de encadenar peticiones al usuario e interactuar de una manera más natural a como se haría simplemente con los cuestionarios. Además el uso de las funciones externas permite modificar aspectos de una manera más rápida y mejorar la integración con el resto de la plataforma de e-learning, por ejemplo para agregar calificaciones a los ejercicios del simulador.

## **2.4. Aplicaciones web**

### **2.1.1 ¿Qué son?**

Los servicios de la Web 2.0 son aplicaciones web de muchos y diversos tipos. El funcionamiento de las aplicaciones web trata de dejar atrás las páginas estáticas y ofrece contenido dinámicamente respondiendo a las acciones del usuario.

Para ello se utilizan lenguajes del lado del servidor, como PHP o JSP, y lenguajes del lado del cliente, como Javascript, que posibilitan una interacción real con la página. Para trabajar con ellas se utiliza un navegador y conexión a Internet, sin requisitos adicionales de ningún tipo. No es necesaria la instalación de ningún programa adicional.

De esta manera se puede dar un servicio a múltiples usuarios que sólo necesitarán conectarse al sitio donde se encuentre la aplicación web, pudiendo integrar el servicio de esta manera con medios sociales. En cualquier caso la Web está avanzando y cambiando de una forma tan rápida que es difícil saber qué vendrá después. Hoy en día son habituales aplicaciones online que hace unos años parecían inalcanzables (correo web, redes sociales, editores de texto, de imágenes,...). Se empieza a hablar de “la nube” como concepto real, donde las aplicaciones diarias son accedidas de manera remota vía Internet, y la máquina del usuario no es más que un terminal que se conecta a esa nube pero que no requiere almacenar programas o datos más allá de los necesarios para acceder<sup>23</sup>.

Obviamente hay desventajas claras acerca del concepto de “nube”, siendo la primera la existencia de problemas de conexión. Es un punto que deberá resolverse en los próximos años.

### **2.4.2. Ejemplos de aplicaciones web**

En español nos encontramos con un gran número de aplicaciones web de calidad, como pueden ser las ofertadas por los bancos (Bankinter, Caja Madrid, etc.) que tienen, entre otras cosas, una buena usabilidad y un especial énfasis en seguridad.

También están los ejemplos de Menéame.net, infojobs, y muchos otros sitios que ofrecen una gran funcionalidad y son buenos ejemplos de aplicaciones web.

---

<sup>23</sup> Web 2.0. Educause. Disponible en: <http://www.educause.edu/Resources/Web20Applications/154703>  
[Consulta 9-05-09]

En un panorama más internacional se puede destacar Amazon, youtube, last.fm, twitter, picasa, Gmail (de hecho muchas páginas de google, como Docs, Maps, etc.), facebook... la lista es interminable.

Como se puede ver, las aplicaciones web son lo que mueve Internet en estos momentos, las usamos a diario y cada día aparecen más. Mención especial requiere Aula Global, aplicación de la Universidad Carlos III de gestión de contenidos para profesores y alumnos.

### **2.4.3. AJAX**

AJAX, acrónimo de Asynchronous JavaScript And XML (en inglés JavaScript y XML asíncronos) es una técnica de desarrollo web para crear aplicaciones interactivas mediante la combinación de tres tecnologías ya existentes:

- HTML (o XHTML) y hojas de estilos en cascada (CSS) para presentar la información.
- Document Object Model (DOM) para trabajar dinámicamente con los datos.
- XML y XSLT, para intercambiar y manipular datos de manera desincronizada con un servidor web (aunque las aplicaciones AJAX pueden usar otro tipo de tecnologías, incluyendo texto plano, para realizar esta labor).
- Javascript para unirlo todo.

Como el DHTML o LAMP, AJAX no constituye una tecnología en sí, pero es un término que engloba a un grupo de éstas que trabajan conjuntamente.

Las aplicaciones AJAX usan navegadores web que soportan las tecnologías mencionadas más arriba. Entre estos se incluyen Mozilla, Firefox, Internet Explorer, Opera, Konqueror y Safari (siempre que no se trate de versiones obsoletas).

AJAX representa un nuevo enfoque en la usabilidad web, y ofrece la posibilidad de customizar y presentar interfaces mucho más atractivas para las aplicaciones web. Su funcionamiento asíncrono es la parte más importante de la técnica AJAX, puesto que no requiere cargar de nuevo la página web para actualizar o mostrar nuevo contenido.

Entre los mayores impulsores de AJAX se encuentran empresas como Google, que lo



usa activamente en muchos de sus productos, Flickr, Amazon y otros grandes sitios de Internet. Al mismo tiempo está siendo adaptado por empresas y organizaciones más pequeñas, puesto que el aspecto y la experiencia de cara al usuario final del sitio web mejora enormemente.

Se han creado muchas librerías Javascript que integran la funcionalidad de AJAX en su código, como Prototype<sup>24</sup> que probablemente sea la más conocida, y si repasamos las páginas web que usamos habitualmente podremos encontrar que en varias de ellas se usa AJAX activamente. Está totalmente integrado en la Web 2.0 y su validez a la hora de crear aplicaciones web está fuera de toda duda<sup>25</sup>.

### ► UTILIDAD DE AJAX<sup>26</sup>

#### 1. Basado en los estándares abiertos.

Ajax está formado por las tecnologías Javascript, html, xml, css, y XML HTTP Request Object, siendo este último el unico que "no es" estándar pero es soportado por los navegadores más utilizados de internet como son los basados en Mozilla, Internet Explorer, safari y opera.

#### 2. Usabilidad.

Permite a las páginas hacer una pequeña petición de datos al servidor y recibirla sin necesidad de cargarla página entera. El incremento de las actualizaciones "on the fly" elimina el tener que refrescar el navegador, algo bastante apreciado a la hora de operar en una aplicación web.

#### 3. Válido en cualquier plataforma y navegador.

Internet Explorer, los basados en Mozilla y Firefox son los que se llevan la palma en el mercado de internet y además son los navegadores en los que es mas fácil programar aplicaciones Web AJAX, pero ahora es posible construir aplicaciones web basadas en AJAX para que funcionen en los navegadores mas modernos. Es una de las razones

---

<sup>24</sup> Prototype es un Framework de trabajo con Javascript que pretende facilitar el desarrollo de aplicaciones web. Disponible en: <http://www.prototypejs.org/> [Consulta 4-05-09]

<sup>25</sup> Puntos de vista sobre AJAX defendidos por Jesse James Garrett, reconocido diseñador de productos tecnológicos [<http://adaptivepath.com/aboutus/jig.php>], de cuyos artículos se hace eco también Mariano Amartino, consultor especialista en Internet y Social Media . "AJAX un nuevo acercamiento a Aplicaciones Web". Disponible en <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php> [Consulta 4-05-09]

<sup>26</sup> MELLADO, Javier. Diez razones para usar AJAX. *AJAX hispano*. 17 de enero de 2006. Disponible en <http://www.ajaxhispano.com/diez-razones-para-usar-AJAX.html> [Consulta 10-05-09]

mas importantes por las que AJAX se ha vuelto tan popular. Aunque si bien muchos desarrolladores sabían que era posible usarse años atrás con Internet Explorer, no era viable realizarse. Ahora ya es posible su avance gracias a Mozilla y Firefox.

#### 4. Beneficia las aplicaciones web.

AJAX es la cara del presente en las aplicaciones web - las aplicaciones web conllevan ciertos beneficios sobre las aplicaciones sobre escritorio (aplicaciones que dependan de un sistema operativo, librerías, lo que entendemos por programas compilados). Esto incluye un menor coste de creación, facilidad de soporte y mantenimiento, menores tiempos a la hora de desarrollarlas, y sin necesidad de instalaciones; éstas son algunas de los beneficios que han llevado a las empresas y usuarios el adoptar aplicaciones web desde mediados de los 90. AJAX solo ayudará a las aplicaciones web a mejorar y conseguir un mejor resultado de cara al usuario final.

#### 5. No es difícil su utilización.

Porque AJAX esta basada en los estándares que han sido utilizados durante muchos años, muchos desarrolladores web han tenido que utilizar las tecnologías que las aplicaciones AJAX requieren. Esto significa que no es un gran esfuerzo el aprendizaje de los desarrolladores el pasar de un simple código HTML y aplicaciones web a una potente aplicación AJAX. También significa que los desarrolladores pueden actualizar poco a poco las interfaces de usuario hacia unas interfaces con AJAX; no necesita una re-escritura de la aplicación entera, se puede hacer incrementalmente.

#### 6. Compatible con Flash.

Muchos desarrolladores tienen serias dudas sobre usar Flash o AJAX. Definitivamente hay ventajas y desventajas en ambas tecnologías según la situación que se dé pero también hay muchas posibilidades y muy buenas para que ambas funcionen en conjunto.

#### 7. Adoptado por los "gordos" de la tecnología web.

La difusión de AJAX en los líderes de la industria de internet prueba que el mercado acepta y valida el uso de esta tecnología. Todo el mundo esta migrando hacia AJAX incluyendo Google, Yahoo, Amazon, Microsoft (por nombrar unas pocos). Google Maps fue lo que captó la atención de los desarrolladores web. Cuando empezaron a investigar como Google era capaz de llevar esa increíble herramienta dentro de un

navegador sin necesidad de ningún tipo de plug-in, encontraron que AJAX estaba detrás del tema.

#### 8. Web 2.0.

El movimiento Web 2.0 está cada vez más en auge y dando quebraderos de cabeza de muchos programadores, usuarios, y vendedores. Esto está ayudando la adopción de AJAX. Las interfaces de AJAX son un componente clave de muchas de las aplicaciones Web 2.0, como puede ser BackPack (un organizador de disco online en entorno Web) y Google Maps. Afortunadamente gracias a todo lo que se le está dando, acelerará la adopción de AJAX y los beneficios de su uso lo mantendrá en escena. Una de las claves principales de Web 2.0 es el usar la red como plataforma para el desarrollo de aplicaciones, en vez de simples páginas web. Siendo importante la iteración de los usuarios con la aplicación en sí.

#### 9. Es independiente del tipo de tecnología de servidor que se utilice.

Así como AJAX funciona en cualquier navegador, es perfectamente compatible con cualquier tipo de servidor estándar y lenguaje de programación Web. PHP, ASP, ASP.Net, Perl, JSP, Cold Fusion. El ser completamente compatible el desarrollo en estas tecnologías ha ayudado a AJAX a que vaya cada vez más en auge.

#### 10. Mejora la estética de la web.

Con AJAX se puede interactuar la imaginación del desarrollador con la usabilidad de una aplicación web de forma que se pueda realizar una aplicación que si no estuviera dentro de un navegador, podría pasar por una aplicación normal de escritorio.

### ► DESVENTAJAS DE AJAX

A pesar de todo lo bueno que ofrece AJAX existen ciertas desventajas a tener en cuenta cuando se trabaja con esta técnica.

La más clara es la dificultad añadida al trabajar con esta técnica, puesto que frente a los hipervínculos para mostrar otra página se requiere una estructura diferente y más compleja para trabajar con AJAX.

Además se corre el riesgo de usarlo indebidamente, en lugares donde no es imprescindible y se retrasa el desarrollo por un mal planteamiento.

Por último se añaden requisitos a la navegación. A pesar de que la mayoría de navegadores actuales pueden trabajar con AJAX hay un pequeño porcentaje de usuarios que no podrían acceder. También es necesario tener en cuenta la carga del servidor y el tráfico de la red, puesto que si la navegación normal es lenta, usando AJAX, aunque se requiera menos tráfico, la respuesta debería ser más inmediata.

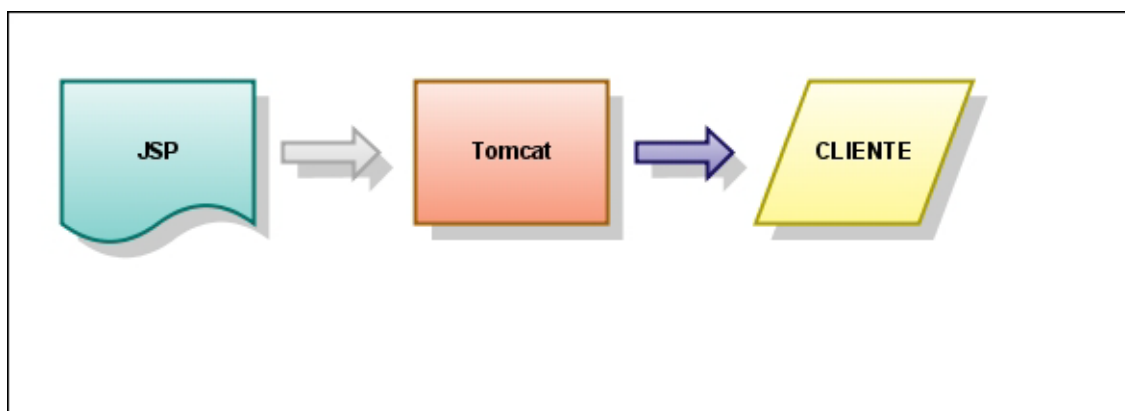
#### 2.4.4. JSP

JavaServer Pages (JSP) es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo<sup>27</sup>.

Esta tecnología es un desarrollo de la compañía Sun Microsystems. La especificación JSP 1.2 fue la primera que se liberó y en la actualidad está disponible la especificación JSP 2.1.

Las JSP's permiten la utilización de código Java mediante scripts embebidos en el HTML. Puede considerarse como una manera alternativa, y simplificada, de construir servlets. Es por ello que una página JSP puede hacer todo lo que un servlet puede hacer, y viceversa. Cada versión de la especificación de JSP está fuertemente vinculada a una versión en particular de la especificación de servlets<sup>28</sup>.

El funcionamiento general de la tecnología JSP es que el Servidor de Aplicaciones interpreta el código contenido en la página JSP para construir el código Java del servlet a generar. Este servlet será el que genere el documento (típicamente HTML) que se presentará en la pantalla del Navegador del usuario.



**Ilustración 2. Funcionamiento básico de JSP**

---

<sup>27</sup> JavaServer Pages Technology. Disponible en <http://java.sun.com/products/jsp/> [Consulta 12-05-09]

<sup>28</sup> Ver tabla relacional en el apartado siguiente: Tomcat.

El rendimiento de una página JSP es el mismo que tendría el servidor equivalente, ya que el código es compilado como cualquier otra clase Java. A su vez, la máquina virtual compilará dinámicamente a código de máquina las partes de la aplicación que lo requieran. Esto hace que JSP tenga un buen rendimiento y sea más eficiente que otras tecnologías web que ejecutan el código de una manera puramente interpretada (por ejemplo PHP).

La principal ventaja de JSP frente a otros lenguajes es la utilización de Java para la lógica que funciona por debajo. Esto permite separar en niveles las aplicaciones web, dejando la parte encargada de generar el documento HTML en el archivo JSP.

Otra ventaja es que JSP hereda la portabilidad de Java, y es posible ejecutar las aplicaciones en múltiples plataformas sin cambios, gracias a la Máquina Virtual de Java.

#### ► TOMCAT

El proyecto Tomcat comenzó en Sun Microsystems para trabajar en la creación y especificaciones de Servlets y Java Server Pages. Posteriormente el código fue donado por Sun a la Apache Software Foundation en 1999, y la primera versión de Apache que salió a la luz fue la 3.0. Desde entonces un gran número de voluntarios de Sun y numerosas organizaciones han contribuido a su desarrollo. Se han producido muchas mejoras a lo largo de todas sus versiones, y ha conseguido una gran aceptación al igual que una activa y numerosa comunidad<sup>29</sup>.

En 2005 Tomcat pasó a ser un proyecto principal de Apache, abandonando el proyecto Jakarta<sup>30</sup> y pasando a gestionarse de forma independiente.

Apache Tomcat es una implementación open source de las tecnologías Java Servlet y JavaServer Pages. Las especificaciones de dichas tecnologías se desarrollan bajo el Java Community Process. Tomcat se desarrolla en un entorno abierto y participativo ofrecido bajo la licencia Apache Software License<sup>31</sup>. Pretende nutrirse de una colaboración de los mejores desarrolladores del mundo, por lo que invitan a participar a quien quiera unirse al proyecto.

---

<sup>29</sup> Apache Tomcat. The Apache Software Foundation. Disponible en <http://tomcat.apache.org/> [Consulta 12-05-09]

<sup>30</sup> Jakarta es un proyecto de la Apache Software Foundation que agrupa diversas soluciones Open Source de Java. Disponible en <http://jakarta.apache.org/> [Consulta 12-05-09]

<sup>31</sup> Licenses. The Apache Software Foundation. Disponible en: <http://www.apache.org/licenses/> [Consulta 12-05-09]

Las diferentes versiones de Apache Tomcat están disponibles para diferentes versiones de las especificaciones de Servlets y JSP. La correlación está en la tabla siguiente:

Servlet/JSP Spec	Apache Tomcat version
2.5/2.1	6.0.x
2.4/2.0	5.5.x
2.3/1.2	4.1.x
2.2/1.1	3.3.x (archived)

**Ilustración 3. Relación de versiones de Tomcat con especificaciones de Servlet/JSP**

### 2.4.5. PHP

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP<sup>32</sup>.

Lo que distingue a PHP de la tecnología Javascript, la cual se ejecuta en la máquina cliente, es que el código PHP es ejecutado en el servidor. Si tuviésemos un script similar al de nuestro ejemplo en nuestro servidor, el cliente solamente recibiría el resultado de su ejecución en el servidor, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. El servidor web puede ser incluso configurado para que procese todos los archivos HTML con PHP.

Lo mejor de usar PHP es que es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales. No sienta miedo de leer la larga lista de características de PHP, en poco tiempo podrá empezar a escribir sus primeros scripts.

Aunque el desarrollo de PHP está concentrado en la programación de scripts en el lado del servidor, se puede utilizar para muchas otras cosas.

---

<sup>32</sup> Todos los datos y características de PHP han sido obtenidos de los manuales de su página oficial, donde se puede encontrar mucha más información y toda la documentación referente a este lenguaje de programación. Disponible en: <http://www.php.net/manual/es/index.php> [Consulta 13-05-09]

PHP puede hacer cualquier cosa que se pueda hacer con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies. Y esto no es todo, se puede hacer mucho más.

Existen tres campos en los que se usan scripts escritos en PHP.

- Scripts del lado del servidor. Este es el campo más tradicional y el principal foco de trabajo. Se necesitan tres cosas para que esto funcione. El intérprete PHP (CGI ó módulo), un servidor web y un navegador. Es necesario correr el servidor web con PHP instalado. El resultado del programa PHP se puede obtener a través del navegador, conectándose con el servidor web.
- Scripts en la línea de comandos. Puede crear un script PHP y correrlo sin ningún servidor web o navegador. Solamente necesita el intérprete PHP para usarlo de esta manera. Este tipo de uso es ideal para scripts ejecutados regularmente desde cron (en \*nix o Linux) o el Planificador de tareas (en Windows). Estos scripts también pueden ser usados para tareas simples de procesamiento de texto
- Escribir aplicaciones de interfaz gráfica. Probablemente PHP no sea el lenguaje más apropiado para escribir aplicaciones gráficas, pero si conoce bien PHP, y quisiera utilizar algunas características avanzadas en programas clientes, puede utilizar PHP-GTK para escribir dichos programas. También es posible escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal

## ► CARACTERÍSTICAS IMPORTANTES DE PHP<sup>33</sup>

### 1) La Comunidad PHP

PHP tiene una comunidad muy grande de desarrolladores. Existen miles de lugares donde se pueden encontrar: documentación, tutoriales, ejemplos de código, foros, etc. Si se tiene un problema con PHP se puede encontrar la respuesta en muchos sitios en donde los usuarios comparten el conocimiento.

### 2) Aprender PHP es fácil

PHP es fácil de aprender comparado con otros lenguajes de programación. El

---

<sup>33</sup> 10 razones para usar PHP. Entrada de blog.: Unijimpe, 25 de septiembre de 2007. Disponible en: <http://blog.unijimpe.net/10-razones-para-usar-php/> [Consulta 13-05-09]

lenguajes es semejante a C ó Java, pues la sintaxis primaria esta basada en Perl. Además tiene semejanzas con Javascript o ActionScript por ejemplo en sus estructuras de control. Otro punto es que PHP tiene librerías especializadas en determinados trabajos por lo cual solo se necesita conocer la sintaxis y aplicarla para lograr buenos resultados.

3) Rendimiento

El rendimiento de PHP es muy bueno, verdaderamente eficiente, utilizando un servidor modesto se pueden atender millones de peticiones al día. Además de ello si es necesario mejorar este rendimiento existen múltiples configuraciones que potencien diferentes aspectos del servidor en función de los requisitos.

4) Bajo Coste

El precio para utilizar PHP es cero, PHP es gratuito y se puede descargar desde [www.php.net](http://www.php.net). Incluso si se contrata un hosting sale mas barato uno con soporte PHP comparado con el que tiene soporte ASP o ASP.NET.

5) Es Open Source, se puede modificar

PHP es Open Source es decir que se tiene acceso al código fuente. Si se desea agregar o modificar algo para obtener un funcionamiento de acuerdo a unas necesidades específicas puede hacerse con total libertad. Esto es la gran diferencia respecto a las aplicaciones comerciales en las cuales sólo queda esperar versiones mejoradas de la empresa desarrolladora. Esto además implica tener acceso al código miles de desarrolladores que detectan bugs y van corrigiendo y mejorando PHP, logrando tener una aplicación más segura y constantemente mejorada.

6) Librerías Incluidas

PHP fue diseñada para trabajar sobre la web, por ello ofrece un conjunto muy amplio de funciones para ser utilizadas en diferentes tareas relacionadas. Se puede conectar con bases de datos, conectar a web services, parsear XML, enviar email, generar PDFs, generar imágenes, etc. Basadas en estas librerías existen clases implementadas para facilitar el trabajo de los desarrolladores. Otro punto es que hay desarrolladores que agregan librerías especializadas para extender las funcionalidades de PHP.

7) Portabilidad



PHP esta disponible para la mayoría de sistemas operativos existentes. Desde Unix, Linux, Microsoft Windows o MAC, entre otros. Una vez desarrollada la aplicación PHP, puede funcionar en cualquiera de estos sistemas operativos sin necesidad de modificar el código gracias al intérprete de la plataforma.

8) Soporte para OOP

La versión 5 de PHP esta diseñada para dar soporte a características de programación orientada a objetos. Características como herencia, métodos y atributos públicos o privados, clases y métodos abstractos, constructores, interfaces y destructores. Si se tienen conocimientos de C++ o Java estas características te serán muy familiares con una sintaxis muy similar.

9) Soporte para gran variedad de Bases de Datos

PHP tiene soporte para conectarse a una gran variedad de base de datos como: MySQL, PostgreSQL, mSQL, Oracle, dbm, FilePro, HyperWave, Informix, InterBase, Sybase entre otras. Las base de datos hacen que una aplicación sea mas robusta y con este soporte cualquier aplicación puede conectarse con facilidad una base de datos existente.

10) Soporte

Si lo que necesitas es soporte, Zend Technologies la empresa que patrocina PHP, ofrece versiones comerciales con todo el soporte que se pueda necesitar.

PHP es la base de gran parte de internet hoy en día. Gracias a los gestores de contenidos basados en él, que permiten montar en muy poco tiempo una web, así como la gran cantidad de aplicaciones que forman la base de grandes entornos (plataformas de creación de blogs, noticias, mensajerías, redes sociales, plataformas de e-learning -CMS, LMCS- ,etc.) hacen de él un lenguaje imprescindible.

### 2.4.6. Comparativa PHP vs JSP

Para hacer una web simple php es la mejor opción, rápido y flexible. Pero si hablamos de proyectos más complejos como por ejemplo lógica de empresa, gestión y comercio electrónico u otras soluciones lo ideal es usar java.

Para poder comparar se necesita tener en cuenta las características y necesidades de la aplicación y de las diferencias entre PHP y JSP, que determinarán las mejores opciones.

Con ambas tecnologías se han desarrollado sistemas muy importantes, lo cual demuestra la potencia de las dos.

Lo aconsejable es que si se necesita desarrollar como lógica del sistema web algo que pueda ser denominado como una aplicación con una complejidad media/alta, es preferible usar Java<sup>34</sup>.

#### **JSP tiene dos ventajas:**

- 1) Permite de forma más intuitiva una separación entre presentación y lógica. Con PHP se puede conseguir usando un buen diseño (quizás utilizando plantillas o templates, uso de XML), pero es menos intuitivo.
- 2) La programación de JSP/Servlet utiliza Java como herramienta lo cual tiene ventajas al ser un entorno más orientado al desarrollo de aplicaciones (las comprobaciones estáticas de tipos, herramientas de desarrollo como netbeans o eclipse, API más genéricas, etc.).

Por su parte, el PHP presenta como ventaja que está más orientado a Web, por lo que su uso es más intuitivo cuando la lógica es sencilla. Realizar cualquier modificación o añadido es muy inmediato.

En resumen: Si es más adecuado realizar la parte dinámica en un lenguaje más típico de desarrollo de aplicaciones como Java el uso de JSP/Servlet sería conveniente. Si la lógica es sencilla el PHP puede ser mejor.

Con esto no se quiere decir que en PHP no se pueda realizar, pero puede suponer

---

<sup>34</sup> Así lo recomienda Javier Vera Gómez en su presentación *Comparativa JSP, PHP, CGI, Servlets, Applet* presentado en el Seminario de Conceptos Avanzados de Sistemas Operativos en la Universidad Politécnica de Cataluña. Disponible en: <http://studies.ac.upc.edu/FIB/CASO/seminaris/2q0102/T8.ppt> [Consulta 15-05-09].

mucho mayor esfuerzo al no ser un lenguaje adecuado para programación más general.

Respecto a ASP.NET, Microsoft ha cambiado con el tiempo su estrategia de desarrollo web. Antes, el ASP tenía la filosofía de un lenguaje orientado a la presentación entre código HTML, con lo que era más equivalente al PHP.

Con la aparición de .NET, el ASP.NET adquiere una filosofía equivalente al JSP en el que se intercala no código sino algo similar a “etiquetas ASP” que se ejecutarán en el servidor.

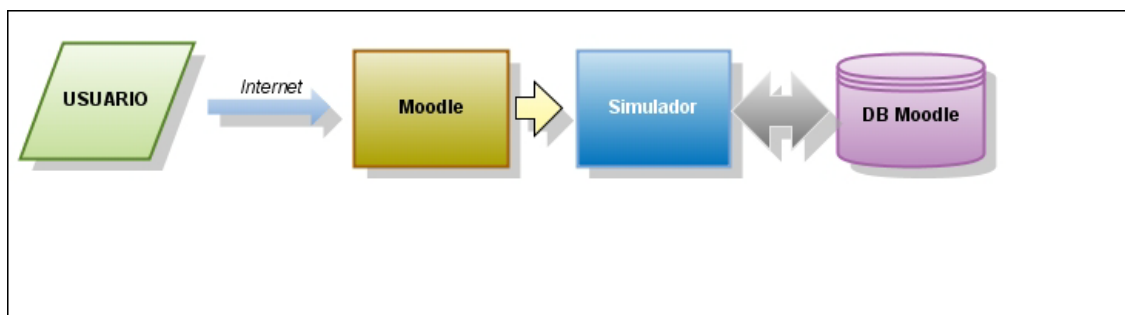
La arquitectura .NET es realmente cómoda, permite la programación en varios lenguajes (c#, asp, e incluso cobol o lisp) y la integración de los resultados. No obstante los test de rendimiento arrojan resultados en contra de .NET en general y de ASP en particular, manteniéndose a la cabeza Apache+PHP. Además existe el problema añadido del coste de los entornos de desarrollo en .NET, licencias y demás, que son algo importante a considerar.

### 3. ARQUITECTURA

Una vez estudiado el estado del arte sobre las tecnologías para el e-learning y la creación de contenidos, habiendo escogido las mejores opciones para el desarrollo del simulador y habiendo visto, en definitiva, todas las posibilidades que ofrecen para cumplir con nuestros objetivos, pasamos en este punto a desarrollar y concretar la arquitectura utilizada para elaborar el proyecto.

Las explicaciones estarán divididas en diferentes puntos para tratar todos los aspectos del proyecto, desde la creación del curso en Moodle hasta el trabajo con el simulador.

Partimos de la siguiente idea general que hemos ido desarrollando:



**Ilustración 4. Base del funcionamiento de la aplicación.**

Un aspecto importante a destacar es que todo el proyecto se ha desarrollado usando software de acceso libre para ofrecer una licencia final lo menos restrictiva posible. En los puntos finales listaremos las aplicaciones usadas y la licencia final propuesta para el proyecto.

Sin más dilación pasamos a explicar el primer punto del trabajo: el estudio previo y los requisitos.

### **3.1. *Diseño del curso***

#### **► VALORES EDUCATIVOS**

En el curso se va a establecer una aproximación gradual en la parte teórica y una base continua y estable en la parte práctica. Al ir ofreciendo unas primeras lecciones sencillas y luego aumentar la complejidad paulatinamente se consigue una mayor concentración y un mejor aprendizaje posterior. A cambio se pierde intensidad en los primeros temas, y consecuentemente imposibilita un aumento del contenido en el temario. Sin embargo creemos que es positivo aumentar la dificultad poco a poco para favorecer la integración del alumno desde el principio. Se podría pensar que es posible que se produzca una falta de interés al principio, pero con un temario claro y específico el alumno sabe hasta dónde se va a profundizar en el curso, tiene la posibilidad de dividir mejor sus esfuerzos y coordinarlos con los temas propuestos para sus diferentes asignaturas u ocupaciones, lo cual es una ventaja importante que ofrece el aprendizaje online.

La parte práctica irá a la par en dificultad con la teoría, puesto que se basará en el temario dado. En apartados posteriores se hablará de los contenidos de la parte práctica al tratarse de puntos necesarios para el proyecto que tienen importancia más allá del curso.

#### **► BASES DE ENSEÑANZA ONLINE**

Se establecerán diferentes recursos para facilitar el devenir del curso, entre ellos un foro y un chat. El foro cumple con la misión de mantener una línea de contacto entre el profesor y los alumnos donde resolver las dudas que surjan y tratar las posibles dificultades inherentes a la docencia. Además es un medio excelente para adelantar las noticias relativas al curso y de interés general.

El chat, por otro lado, ofrece un sitio para que el alumno esté más en contacto con el resto de alumnos. No es tan útil para un profesor como el foro, pero puesto que se ofrece la opción de tenerlo es una buena idea aprovecharla.

Adicionalmente se aprovecharán las opciones características de las plataformas de e-learning para ofrecer al alumno y al profesor un seguimiento constante de los avances en el curso, de las notas, comentarios y otros elementos.

## ► ESTRUCTURAS DE ENSEÑANZA APLICADAS

El curso, dado el material teórico del que se dispone, se estructurará en nueve temas, con uno añadido de introducción. Cada tema tratará un aspecto determinado de Amadeus, siguiendo la evolución gradual explicada anteriormente, hasta cubrir un temario.

Se van a evaluar dos posibilidades para tratar el avance del alumno:

- la primera consiste en establecer cada tema como un elemento cerrado que el profesor va descubriendo a los alumnos en una fecha determinada. Es una opción sencilla que garantiza un seguimiento básico del curso, fácil de implementar en cualquier sistema de e-learning;
- la segunda permitiría al alumno avanzar en el curso según su propio desempeño, mediante un sistema de valoración que cerrase el acceso a temas por encima de la calificación actual del alumno. Es también una buena opción, pero en muchas plataformas no está disponible, o en otras no lo está directamente.

La teoría de cada tema estará incluida en un sólo documento, en formato html, con posibilidad de añadir otro tipo de documentación adicional pero que no entraría en la evaluación. Para este proyecto y para este curso creemos que es mejor centralizar la teoría del tema en un solo documento, más aún tratándose de temas de relativa corta duración como es el caso, en lugar de dividir en pequeños apartados que pudieran desviar la atención de algún punto importante.

Los ejercicios prácticos de cada tema estarán enfocados a garantizar el conocimiento de la teoría. Para eso se dividirán en dos tipos: cuestionarios y simulaciones.

Los cuestionarios constarán de 10 preguntas sencillas relacionadas con la parte teórica del tema. Se tratará de incluir diferentes tipos de preguntas, de elección múltiple, verdadero-falso, rellenar el hueco, etc.

Las simulaciones tratarán la teoría vista hasta el tema actual, pudiendo requerir conocimientos de temas anteriores. Algo lógico si se cuenta con el aprendizaje incremental del que ya hablamos antes. Puesto que las simulaciones son uno de los puntos más interesantes del curso se ofrecerán varios ejercicios por tema calificando únicamente uno de ellos (elegido por el profesor). De esta manera se ofrece al alumno

un entorno seguro de prácticas que le permita afrontar los ejercicios que entren en la calificación con total garantía.

### ► SCORM Y OTROS MÉTODOS DE ESTANDARIZACIÓN

SCORM nos permite integrar el curso en un paquete y hacerlo portable y reutilizable para diferentes plataformas de e-learning. Es una opción interesante dado que cada gestor de contenidos implementa su propio estándar y hace difícil o imposible el intercambio de contenidos. Es un método a tener en cuenta, aunque en principio nos centraremos en la plataforma que finalmente reciba el curso, trataremos de valorar la utilización de SCORM o de otras especificaciones similares.

### 3.1.2. Ejercicios

#### ► UTILIDAD DEL SIMULADOR

La utilidad de los ejercicios de simulación del entorno Amadeus es innegable, pero para integrarlo dentro de un curso, como se requiere en este proyecto, hay que valorar diferentes posibilidades y formas de aproximarse al problema. Es posible que una buena forma de implementar y trabajar con un simulador no fuese adecuada afrontándolo desde la perspectiva de un curso.

En nuestro caso pretendemos centrarnos en la interacción del usuario con un sistema similar a Amadeus, para así facilitar el aprendizaje de las lecciones. Preferimos sacrificar fidelidad con el sistema original a cambio de ganar en sencillez y en facilidad de integración con el simulador, siempre manteniendo el estilo y el funcionamiento acorde con el sistema original y con lo explicado en las lecciones de teoría.

#### ► APLICACIÓN WEB

El simulador se creará como una aplicación web. Deberá funcionar en el rango más amplio de navegadores y adecuarse a una pantalla de terminal de Amadeus lo más fielmente posible. A pesar de la variedad de interfaces para Amadeus existentes en el mercado, se decidió adoptar para el proyecto el modelo sencillo de terminal de texto con fondo azul y fuente en blanco.

La aplicación funcionará tanto con programación del lado del cliente como del servidor,

con el propósito de ser una aplicación ligera y fácil de usar. En puntos posteriores se valorarán las diferentes alternativas para implementar la solución en función de los requisitos establecidos.

### ► VENTAJAS

Las ventajas de utilizar esta tecnología son a priori el acceso desde cualquier lugar con conexión a internet, los reducidos requisitos hardware y software necesarios, la integración con el entorno de e-learning y la modificación, ampliación de funcionalidades o corrección de errores de forma centralizada y transparente al usuario.

La desventaja más clara es la necesidad de una conexión a internet, limitando la práctica offline, pero dada la naturaleza del curso, la utilización de plataformas de e-learning entre otros aspectos, no parece que vaya a suponer un hándicap importante para los estudiantes.

### ► FUNCIONAMIENTO DEL SIMULADOR

Respecto al funcionamiento del simulador se dividirá en dos partes diferenciadas, la interfaz del cliente y la programación en el lado del servidor.

Para que el estudiante acceda a un ejercicio simplemente deberá pulsar un enlace, entonces se abrirá una nueva pantalla del navegador simulando el entorno de Amadeus con un texto que indique el ejercicio a realizar. Según van siendo introducidos los códigos por el alumno se avanzará en el ejercicio. Existirá la posibilidad de pedir ayuda a cada paso, y al completar el paso final requerido se añadirá la nota automáticamente al usuario dentro de la plataforma de e-learning.

Por el otro lado se implementará un sistema que ofrezca al usuario todo lo expuesto en el anterior párrafo. Los ejercicios se establecerán mediante lenguaje XML para permitir una mayor flexibilidad tanto en la creación como en la modificación de los mismos.

Dentro de cada ejercicio XML se ofrecerá la posibilidad de aceptar funciones externas que influyan tanto en la recepción de los datos introducidos por el usuario como en la salida por pantalla. De esta manera se tratará de ofrecer un sistema básico que a su



vez sea ampliable añadiendo funciones que se cargarán de forma independiente.

### ► ¿SIMULAR O CREAR EJERCICIOS?

Es una duda razonable que surge al tratar de ofrecer una herramienta como la que estamos proponiendo, y la respuesta no está clara sin un razonamiento previo. En este caso se cree más adecuado ofrecer ejercicios cerrados por dos razones: la primera es que es una aplicación más sencilla que un simulador completo; la segunda es porque para un alumno es más directo enfocarse en un problema concreto que ofrecer la libertad de un simulador más abierto en el que el camino a seguir no esté suficientemente claro.

Respecto a la primera razón las causas son obvias, no es lo mismo diseñar una aplicación que acepte unos determinados archivos de ejercicios que implementar un mini-amadeus, que por reducido que fuera la complejidad sería muy superior, y probablemente el resultado no sería tan diferencialmente superior, al menos en el plano educativo. En cualquier caso, para no cerrar esa interesante puerta al simulador se quiere ofrecer la posibilidad de añadir funciones externas que pudieran añadir esa funcionalidad extra y acercar la aplicación a un simulador más completo.

En cuanto a la segunda razón, ésta es más de naturaleza pedagógica que una barrera tecnológica. Ante la libertad de decisión que otorgaría un simulador muchos alumnos agradecerían enormemente el esfuerzo dedicado a ofrecerlo, pero muchos otros se podrían sentir perdidos ante la falta de una guía más estricta. Lo ideal sería un término medio entre los dos extremos, y es lo que trataremos de ofrecer en el curso. No obstante, ante la duda creemos que es preferible ser estrictos con lo que se le pide al alumno para conseguir una mayor y más rápida aceptación del curso, además de ofrecer una mejor preparación.

### ► ELECCIÓN DE APLICACIÓN WEB

Entre las opciones disponibles para escoger la alternativa vamos a decidir, ahora, las que usaremos en el proyecto. En próximas páginas hablaremos de las razones que nos ayudan a elegir entre una u otra plataforma, ahora vamos a explicar el proceso de trabajo que será necesario para organizar la aplicación web.

Para el código del lado del servidor vamos a montar un servidor Tomcat (la versión

disponible en la Universidad es la 5.5.23) para servir páginas JSP. Estas páginas van a servir de interfaz de entrada/salida para interactuar con las clases Java que llevarán la parte más pesada del programa.

En el lado del cliente usaremos javascript y css sobre páginas xhtml para dar el aspecto de terminal a la aplicación. Implementando un sencillo formulario con el código y el estilo adecuados se puede conseguir simular la interfaz de un terminal de Amadeus funcional acorde con las especificaciones requeridas. Usado conjuntamente con AJAX se obtiene una funcionalidad dinámica y directa.

El proceso es el siguiente: la página cargada por el usuario, con el estilo aplicado convenientemente, recoge el texto introducido en un campo y hace una llamada asíncrona a una página JSP, donde se procesarán los datos recibidos y se devolverá una respuesta en código xhtml para mostrar por pantalla. La respuesta se obtendrá a través de los archivos XML de cada ejercicio y de las funciones especificadas en ellos. Se contempla el uso de sesiones JSP para almacenar información específica relacionada con datos de carga de ejercicios o de validación de usuarios.

Con esta información vamos a proceder a explicar a partir de aquí el por qué de nuestras decisiones y la forma en que será implementada la solución escogida. Toda la aplicación estará disponible para interactuar con Moodle, directamente o a través de MySQL, para facilitar la evaluación de los alumnos.

Además se pretende que sea accesible y modular, de modo que acepte nuevas funcionalidades como se ha descrito anteriormente. Esto permitirá el uso de esta aplicación en otros proyectos y entornos de desarrollo con mínimos cambios requeridos. En principio simplemente cambiando las opciones establecidas en los XML y el estilo e implementación de la página (que en nuestro proyecto hace las funciones de terminal simulada) se estaría ofreciendo una funcionalidad diferente. Aquí entraría la versatilidad de las funciones para añadir extras a la aplicación.

## 3.2. Tecnología

### 3.2.1 Requisitos

#### ► REQUISITOS CLIENTE

Los requisitos básicos del usuario para participar el curso son conexión a internet y un navegador Web. En este caso los requisitos vienen determinados por los establecidos para el navegador web.

Los requisitos establecidos por Mozilla Firefox para el navegador son:

#### Requisitos del sistema mínimos

Procesador a 233 Mhz

64 MB de RAM

50 MB de espacio libre en disco

Microsoft Windows 98

#### Mozilla recomienda

Procesador a 500 Mhz

256 MB de RAM

100 MB de espacio libre en disco

Microsoft Windows XP

En este caso para el SO Windows, para Linux serían los mismos a diferencia del Sistema Operativo y para Mac valdría con cualquiera equivalente.

Hemos escogido Firefox por ser uno de los navegadores con mayor penetración en el mercado junto con Internet Explorer de Microsoft, y se puede tomar como base. De cualquier manera los requisitos son suficientemente bajos como para que los cumpla cualquier ordenador de hace unos años. Es una de las ventajas que supone el trabajar con aplicaciones web, que los recursos son completamente genéricos y comprobables.

#### ► REQUISITOS DEL SERVIDOR

Respecto al servidor se necesita una instalación con Apache, PHP, MySQL y Tomcat. Dada la naturaleza de las plataformas online estos requisitos podrían cumplirse en un solo ordenador o en varios, es decir, se podría usar un servidor para Apache+PHP, otro para MySQL y uno más para Tomcat.

Apache no tiene unos requisitos específicos tanto en cuanto lo único que hace es funcionar como un servicio adicional del SO sirviendo peticiones web. Un mejor procesador o una mayor memoria mejorarán el funcionamiento en situaciones de

mayor carga, pero no es algo contabilizado en requisitos si no algo necesario de comprobar en pruebas y test de stress del servidor. Junto a Apache tenemos PHP, sin requisitos adicionales más allá de los ya descritos.

Para MySQL los mínimos exigidos son similares a Apache, con la diferencia de que aquí se tiene más en cuenta el espacio necesario para almacenar los datos, recomendándose unos 200MB como mínimo.

Tomcat, del mismo modo que las aplicaciones anteriores, no tiene unos requisitos definidos, dependiendo de la carga de trabajo que vaya a tener el servidor. Por añadido se pueden establecer los requisitos del sistema para funcionar con Java 5.0, que en este caso serían, para una plataforma Windows, 128MB de RAM y alrededor de 100MB de espacio en disco.

Vemos que en general los requisitos del servidor irán en función de la carga que reciba. No sería necesario realizar tests adicionales puesto que a menos que se trabaje con una máquina con bastante antigüedad no habrá problemas de rendimiento.

### **3.2.2. Apache Tomcat + MySQL**

Para tener configurado el servidor en el menor tiempo posible se pueden usar paquetes de aplicaciones preparados para una instalación completa de Apache+PHP+MySQL (conocidos como AMP). Hay un gran número de opciones, ofrecidas incluso por la Apache Foundation, que facilitan enormemente la instalación y puesta en marcha de un servidor.

Tomcat se instalará aparte. Aunque hay versiones de paquetes AMP con Tomcat, creo que es preferible instalarlos de manera independiente ya que Tomcat lo usaremos sólo para servir páginas JSP, aunque podría funcionar de manera análoga a Apache.

No trataremos más en profundidad las diferentes configuraciones porque dependen en gran medida del equipo disponible y del uso que se le vaya a dar. Hay guías de configuración y optimización en las páginas oficiales de cada aplicación, además de docenas de documentos creados por usuarios.

Una gran ventaja de trabajar con estas aplicaciones es que son muy usadas en todo el mundo, están muy probadas y cualquier situación se puede resolver rápidamente gracias a la gran comunidad de usuarios que tienen por detrás. No ocurre lo mismo con otras aplicaciones propietarias donde el mejor recurso suelen ser los foros

oficiales.

Partamos entonces de la base de que con la configuración por defecto podemos trabajar sin problemas, instalar moodle, acceder a la web e instalar la aplicación del simulador, como veremos en los próximos apartados.

### **3.3. Moodle**

#### **3.3.1. Instalación**

Tras la instalación del servidor, configurado correctamente con Apache+PHP+MySQL, pasamos a la instalación de Moodle.

Es un proceso muy sencillo que con un asistente te guiará paso a paso hasta que la plataforma de e-learning esté funcionando y lista para empezar a trabajar. Como es habitual existen varias configuraciones que pueden ser tenidas en cuenta para mejorar el rendimiento, la seguridad y otros factores dependiendo del uso que vayamos a hacer de Moodle.

En nuestro caso el sistema está configurado de acuerdo a las necesidades de la universidad, por lo que partimos de una base sólida, ya que ha demostrado su solvencia durante bastante tiempo, y no tenemos que preocuparnos de la instalación y/o configuración.

Para instalar la aplicación del simulador si que necesitaremos datos de la instalación de Moodle, pero esto ya lo veremos más adelante cuando tratemos el tema del simulador.

#### **3.3.2. Estructura del curso**

Ahora que ya se tiene montado el servidor con Moodle corriendo tenemos que diseñar el curso. Esto es algo más que ir sumando lecciones y ejercicios, hay que pensar en los contenidos que se quieren ofrecer, la manera de presentarlos y el tipo de evaluación.

Nuestro proyecto tiene definidos varios puntos que establecen de antemano cómo va a ser el curso y qué se puede esperar del resultado final. Partimos de que se trata de un curso de iniciación a Amadeus, por lo que el curso se compondrá de lecciones que traten puntos básicos de dicho sistema. Tenemos además dos tipos de ejercicios preparados, unos en forma de cuestionario y otros en forma de simulador de Amadeus, por lo que en este aspecto también está decidido.

Como se puede ver, se tienen los pilares del curso preparados. Ahora vamos a decidir cómo presentarlo y cómo evaluar.

### 3.3.3. Presentación del curso

El curso se dividirá en temas independientes. No se ha planteado la limitación de temas a fechas concretas ni la limitación de contenido en base a la puntuación anterior. Moodle no permite hacer esto directamente, si no a partir del elemento “Lección”, que otorga una puntuación global a cada alumno y a su vez cada elemento puede requerir una puntuación mínima para mostrarse. Es sin duda un método interesante, pero hemos considerado un sistema más sencillo para nuestro curso.

Estará formado por diez temas, primero uno de introducción y luego nueve de contenidos. Cada tema a partir del primero, a su vez, contará con un ejercicio en forma de cuestionario repasando los puntos importantes del tema, y uno o varios ejercicios de simulación.

Para evaluar cada tema se tendrá la nota del cuestionario y la de uno de los ejercicios de simulación. En el caso de que hubiera varios ejercicios sólo habrá uno calificable, los demás son prácticas para reforzar ciertos aspectos de la teoría y para preparar el ejercicio evaluable. La nota de los cuestionarios va del 0 al 10 y la del simulador puede ser apto o no apto, que se considera un 10 y un 0 respectivamente. Vemos que con completar correctamente un ejercicio de simulación ya se tiene la mitad del tema aprobado. En cualquier caso esto es así porque el simulador usa la función de poner la nota de esta manera. Ya veremos cuando llegue el momento de hablar del simulador cómo se tratan las funciones, incluida la de poner nota.

Tenemos, entonces, diez temas, el primero de introducción sin ejercicios, y los otros nueve con dos valoraciones cada uno. El hecho de que la calificación del ejercicio de simulación sea binaria es un factor que refuerza la importancia del simulador, ya que creemos que tiene ventajas frente a otro tipo de ejercicios al ofrecer una situación que requiere de varias acciones para resolverse, frente a preguntas sueltas más específicas. Además con el simulador se pretende crear situaciones realistas que ayuden a familiarizarse con el sistema Amadeus, cosa que de otra manera se intuye más complicada.

Cada tema estará dividido en distintos apartados que tratarán los diferentes puntos propuestos. El desarrollo de cada punto estará apoyado por imágenes y ejemplos a modo de capturas de pantalla para ilustrar el funcionamiento de Amadeus y facilitar la comprensión de las partes más complejas. Los códigos de transacción, el método mediante el cual funciona el sistema Amadeus, estarán adecuadamente señalados y

se adjuntará un resumen de referencia rápida al final de cada tema, para facilitar en lo posible la búsqueda de un código específico o la solución de una situación dada. Adicionalmente se ofrecerá información, a modo de notas, de diferentes propuestas o curiosidades dentro del mundo de Amadeus.

Se ha intentado que cada tema ofrezca un contenido completo, fácil de seguir y útil, además de servir como Introducción a Amadeus, que es lo que se pretende en conjunto.

### **3.3.4. Carga lectiva**

Respecto a los tiempos, estamos hablando de un curso que se va a ofrecer como parte de una asignatura en un cuatrimestre, por lo que hay que aprovechar bien el tiempo.

Al ser online tiene la ventaja de que el tiempo que se le dedica es el que quiera el alumno, pero en general la duración de todos los temas se estima bastante parecida, y es algo que hay que tener en cuenta para planificar tiempos de estudio y de clase.

Cada tema tiene prevista una duración de una hora, que es lo indicado para una clase en un aula informática o para dedicarlo al curso fuera de la Universidad.

Esta hora se divide en unos 15 minutos para leer y comprender la teoría, unos 5~10 minutos para el cuestionario del tema y alrededor de 15~20 minutos para los ejercicios de simulación. Sumando los tiempos tenemos alrededor de 40 ó 45 minutos, que unido a las interrupciones lógicas de la clase, consultas y demás, cumplen el tiempo establecido.

Las clases, presumiblemente, serían una por semana, lo que, quitando el tema de introducción, que no dura esa hora de la que hablábamos, tenemos nueve horas lectivas para nueve semanas de curso, que se acoplan muy bien a lo esperado.

Hay que añadir que la dificultad del curso es ascendente, ya que a medida que se van sucediendo las lecciones se va trabajando con más teoría, lo que en los ejercicios se traduce en más códigos a recordar (o a tener a mano), por lo que no estamos hablando de un curso que se pueda seguir puntualmente una hora, un día a la semana, si no que requiere un seguimiento más cercano, un repaso de lecciones anteriores, tal vez realizar unos apuntes o resúmenes con lo necesario para superar los ejercicios.



En definitiva, aunque la carga lectiva no es alta, se pretende que no sea sencillo en exceso y que requiera la suficiente atención por parte del alumno como para servir como una introducción real al sistema Amadeus.

### 3.3.5. Ejercicios

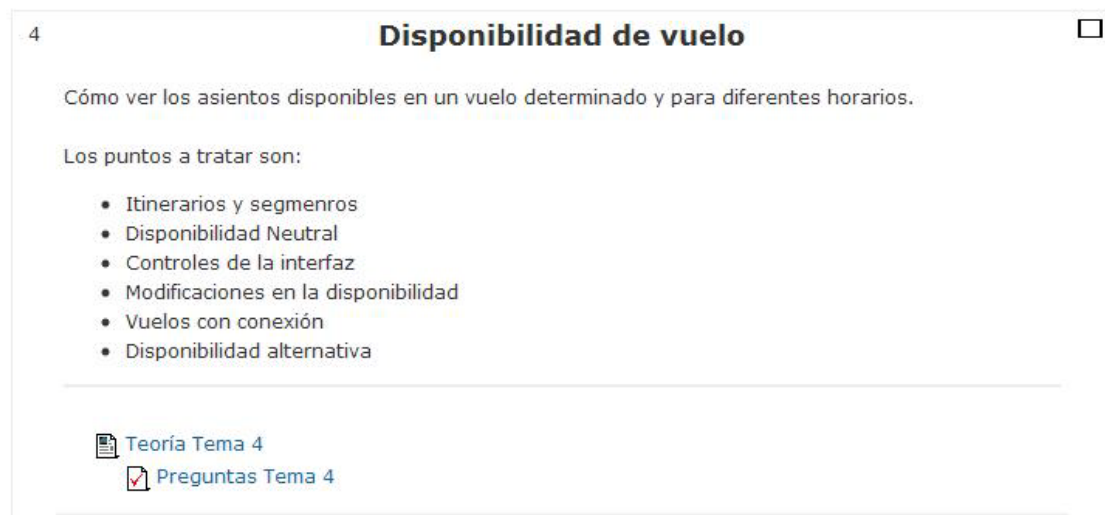
Como ya hemos hablado en apartados anteriores, los ejercicios estarán divididos en dos partes: cuestionarios y simulador.

Los cuestionarios estarán creados en formato GIFT de Moodle. Este formato nos permite poner preguntas de muchos tipos distintos dependiendo de cómo queramos enfocar la teoría, por lo que lo consideramos tremendamente útil.

Los ejercicios del simulador están planteados para tratar todos los puntos del tema al que correspondan, además de poder requerir conocimientos explicados en temas anteriores. En apartados posteriores veremos cómo se crean estos ejercicios, cómo se evalúan y en general cómo se trabaja con el simulador.

#### ► CUESTIONARIOS GIFT

Los ejercicios de cuestionarios en formato GIFT están completamente integrados en Moodle, forman parte de la plataforma y son claros para el usuario.



**Ilustración 5. Presentación de un tema del curso.**

Nosotros nos hemos limitado en los contenidos a preguntar del tema al que corresponde el ejercicio, dejando los contenidos recopilados para los otros tipos de ejercicios.

The screenshot displays a questionnaire interface with four questions, each in a separate box. Each box includes a question number, a question text, a score indicator, and a 'Submit' button.

- Question 1:** ¿Cuales son los GDS más importantes? (Punto/s: --/1). Options: a. Sabre y Galileo, b. Amadeus y Sabre, c. Worldspan y Galileo.
- Question 2:** En Amadeus sólo existe un tipo de acceso a las reservas ofrecidas por las líneas aéreas (Punto/s: --/1). Options: a. falso, b. verdadero.
- Question 3:** Escribe el comando en la calculadora para multiplicar 126 por 3: \_\_\_\_\_. (Punto/s: --/1). Response field: \_\_\_\_\_.
- Question 4:** Introduce el comando para mostrar el día actual: \_\_\_\_\_. (Punto/s: --/1). Response field: \_\_\_\_\_.

**Ilustración 6. Ejemplo de cuestionario.**

Como se ve en la pantalla las preguntas son directas, requiriendo o bien escoger la adecuada en formato de respuesta múltiple o rellenar el hueco.

Entre todas se pueden encontrar además de dos categorías diferenciadas: de código o de teoría. Las de código son preguntas que requieren conocimientos de los códigos de transacción de Amadeus, pidiendo el exacto para acertar. Las de teoría son todas las que no piden estrictamente un código, y suelen referirse a frases o párrafos explicativos dentro de cada lección.

Cada ejercicio constará de 10 preguntas escogidas al azar entre varias (pueden ser sólo 10 ó alrededor de 15) y ordenadas también aleatoriamente. No todos los alumnos tendrán las mismas preguntas ni estarán en el mismo orden. En general son preguntas sencillas y no supondrá una diferencia tener unas u otras, especialmente dada la naturaleza del curso online, en el que se puede tener la teoría delante mientras se contestan.

El profesor puede decidir imponer un límite a estos ejercicios, bloquear la pantalla

(aunque al trabajar desde un navegador no está recomendado por su posible ineffectividad) o utilizar otros mecanismos que ofrece Moodle. Nosotros preferimos dejarlo así y que el profesor asignado al curso tenga libertad para modificarlo si lo cree necesario.

## ► SIMULADOR

Los ejercicios del simulador son más complejos de integrar en Moodle, puesto que trabajamos con una plataforma externa (Tomcat) y nuestra intención es que las calificaciones se integren en Moodle para facilitar la labor docente. Lo ideal sería crear un módulo adicional de Moodle que permitiera crear un ejercicio de simulación y así maximizar la integración con la plataforma, pero esto requeriría un trabajo y un tiempo adicional y por lo tanto quedará para las líneas futuras.

Nosotros hemos adoptado una solución a medio camino y nos valemos de módulos ya creados para ofrecer una funcionalidad similar. En este caso hemos escogido el módulo de Ejercicio Offline, que nos permite realizar justo lo que necesitamos.

Cada tema tendrá asociado uno de estos ejercicios. Pinchando en él nos lleva a una página en la que se nos informa de dónde nos encontramos. Aquí añadimos los enlaces tanto a los ejercicios de práctica como al ejercicio calificable. Puede ser preferible añadir un enlace a una página servida directamente desde Tomcat, para así facilitar el cambio en caso de ser necesario (teniendo acceso a Tomcat puede ser más sencillo modificar algo o añadir ejercicios nuevos que cambiarlo en Moodle).

1


## Presentación

En esta lección nos familiarizaremos con los elementos del curso e introduciremos los temas que vamos a ver durante el mismo.

Los puntos a tratar son:

- Bienvenida
- Los códigos de transacción
- Los ejercicios

---

 [Teoría Tema 1](#)

2


## Introducción a AMADEUS

Primer vistazo al sistema AMADEUS y toma de contacto con los ejercicios. Ejercicios básicos en el simulador.


Los puntos a tratar son:

- Los Sistemas de Distribución Global
- El sistema Amadeus
- El terminal
- Sign In
- Función calculadora

---

 [Teoría Tema 2](#)

 [Preguntas Tema 2](#)

 [Ejercicios simulador tema 2](#)

**Ilustración 7. Los primeros temas del curso en Moodle**

Más adelante hablaremos de cómo funcionan los ejercicios del simulador, cómo se crean y modifican, y de todos los aspectos necesarios para trabajar con ellos.

### 3.4. Curso HTML/ CSS

Para la teoría se han creado páginas web con contenidos organizados, como ya comentamos en apartados anteriores. Para ello se ha utilizado HTML y CSS en el diseño y maquetación de las páginas web.

#### 3.4.1. Diseño

Diseño de la parte teórica de los temas, primero en Moodle, con la presentación, luego en la propia lección.

Puesto que las páginas web tienen el propósito de ofrecer el contenido de la mejor forma posible, hemos preparado un diseño sencillo en dos columnas, manteniendo una a la izquierda con enlaces a los distintos apartados de teoría y la otra con todo el contenido. A su vez todo esto está centrado en la página para centrar la atención del alumno.



**Ilustración 8. Imagen del interior de un tema (detalle de códigos de transacción)**

Los apartados de teoría están adecuadamente separados usando diferentes colores y tonos, recuadrados para facilitar el seguimiento y mantener la atención en los aspectos importantes.

Del mismo modo existen más elementos diferenciados para representar capturas de pantalla, códigos, y otros espacios que refuerzan y apoyan la idea expresada en los grandes bloques de teoría. Se ha trabajado para ofrecer un contenido muy gráfico y visual, dentro de las posibilidades del temario, y así amenizar el transcurrir del curso manteniendo el interés.

### Neutral Availability

La disponibilidad neutral (AN) es la que usa por defecto el sistema. El término Neutral designa que no existen preferencias en cuanto a hora de llegada, hora de salida, o cualquier otro factor, para lo que se usaría otro comando como veremos más adelante. La sintaxis del comando es la siguiente:

**AN[FECHA DE SALIDA][PAR DE CIUDADES][HORA SALIDA(opcional)]**

Si se incluye la hora se mostrarán los vuelos que salgan entre una hora antes y la hora especificada. Por ejemplo, queremos ver la disponibilidad de vuelos para el 26 de Mayo de Madrid a Múnich sobre las 9 de la mañana:

```
AN26MAYMADMUC0900
```

Y se mostraría la siguiente salida:

```
** AMADEUS AVAILABILITY - AN **              75 SA 26MAY 0900
1  IB6223 C9 M9 T9 K9 V9 L9 Q9 MAD 4 MUC 1 0825 1130 0/320 03:05
2  AF1564 C9 M9 T9 K9 V9          MAD 4 MUC 2 0900 1215 0/737 03:15
3  LH7092 C9 M9 T9 K9 V9 L9 Q9 MAD 2 MUC 1 0845 1130 0/310 02:45
```

Vamos a identificar los elementos de esta pantalla. Es importante acostumbrarse a este formato, así de un vistazo podremos obtener una gran cantidad de información. Separaremos los elementos en pantalla y veremos las distintas posibilidades en cada uno de ellos.

La primera línea indica la operación realizada, AN, y unos datos relativos al día y hora de la búsqueda (como todo en Amadeus las fechas también están en inglés).

**Ilustración 9. Imagen del interior de un tema (detalle de pantallas del terminal)**

### 3.4.2. Estándares Web

Se han seguido las líneas marcadas por los estándares web más representativos, para así facilitar el trabajo desde cualquier entorno, navegador o Sistema Operativo. Moodle ya ofrece de por sí una buena compatibilidad con un amplio abanico de posibilidades de configuración, no obstante nosotros hemos establecido la necesidad de mantener los estándares en los documentos HTML que mostramos, puesto que están preparados para el usuario como un recurso importante y una mala visualización sería un grave inconveniente para todo el curso.

Por esto hemos trabajado para mantener un estándar en las hojas de estilo en cascada (CSS) en su versión 2.1 y para el contenido en XHTML en su versión 1.1.

Esta versión no admite la etiqueta transicional, por lo que se considera un estándar estricto lo que favorece, a priori, la compatibilidad con diferentes sistemas.

### 3.4.3. Usabilidad

En este caso en concreto no podemos hablar en profundidad de la usabilidad del curso porque en su mayor parte viene dada por Moodle. En este sentido la plataforma de e-learning nos garantiza una gran facilidad de uso y una sencillez que se agradecen, especialmente las primeras veces cuando se inicia por primera vez, ya que la primera toma de contacto suele ser muy positiva.

En todo caso no hemos querido descuidar nuestra parte en cuanto a la usabilidad, y hemos prestado especial atención al diseño de las lecciones. Como ya explicamos más arriba se han usado dos columnas, sirviendo una de menú, se ha centrado el contenido dejando un porcentaje de espacio libre a los lados y se ha usado un esquema de colores sencillo y que a la vez no distraiga la atención del contenido.

## 3.5. Simulador

En este apartado vamos a hablar del simulador, en qué consiste, cómo se ha implementado y cómo se va a usar.

Amadeus funciona, en su modo de terminal de texto, usando códigos de transacción. Dichos códigos siguen un esquema similar a este:

El código de transacción se muestra en un recuadro con fondo azul claro. El texto es "SS[número\_de\_plazas][clase\_de\_las\_plazas][número\_de\_oferta]/PE". Las partes entre corchetes están coloreadas: "número\_de\_plazas" es verde, "clase\_de\_las\_plazas" es azul y "número\_de\_oferta" es rojo. "SS" y "/PE" son de color negro.

```
SS[número_de_plazas][clase_de_las_plazas][número_de_oferta]/PE
```

**Ilustración 10. Ejemplo de código de transacción.**

El código suele ser de dos o tres caracteres, y los argumentos son de tipos muy variados, pueden ser códigos IATA (por ejemplo para los aeropuertos se usan códigos IATA de tres letras, que son unívocos para cada aeropuerto o ciudad), pueden ser fechas, horas o incluso texto libre.

Hay muchas peculiaridades en el formato de los códigos de Amadeus, pero siempre siguen el mismo esquema básico, por lo que familiarizarse con ellos es útil y necesario para trabajar en Amadeus.



Amadeus también tiene interfaces gráficas que ayudan a mostrar las diferentes opciones de un modo más visual. Principalmente hacen de puente para traducir las acciones del usuario en códigos transaccionales al igual que las respuestas del sistema, pero el modo de terminal no está, ni mucho menos, en desuso.

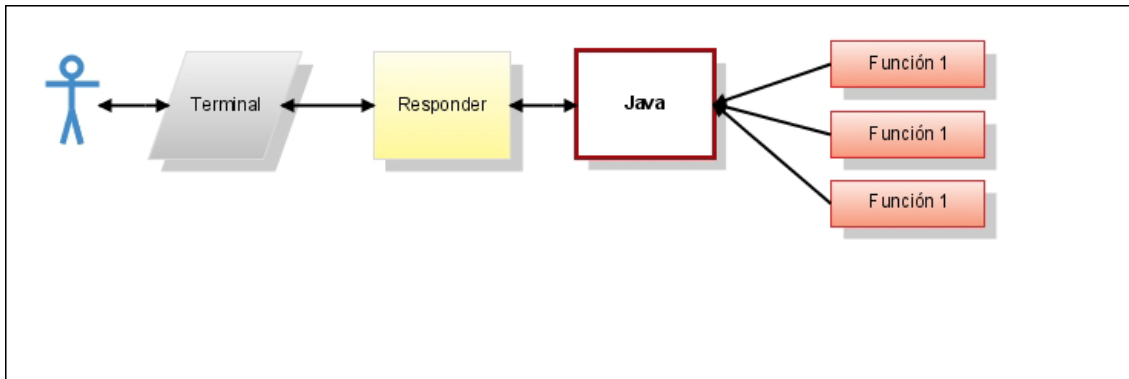
Por esta razón el curso se basa en este formato para explicar la teoría y trabajar con los ejercicios.

Como ya dijimos la idea del simulador proviene de la necesidad de ofrecer ejercicios en los que se pida seguir una serie de pasos, más o menos largos, mediante códigos del sistema Amadeus. Esto es algo que Moodle de por sí no ofrece, aunque se podrían preparar preguntas que requiriesen introducir los mismos códigos en el orden adecuado para llegar al mismo resultado. Sin embargo el simulador nos garantiza dos aspectos fundamentales frente a una lista de preguntas que consideramos fundamentales: la similitud con Amadeus que facilita el familiarizarse con el entorno y la posibilidad de añadir contenido no estático.

En primer lugar la similitud y familiarización con Amadeus se gana desde el momento en que se muestra una pantalla parecida a la ofrecida por Amadeus, en la que el usuario tiene que hacer login, introducir los códigos y observar la respuesta de forma similar a como lo haría en un entorno real.

En segundo lugar, la posibilidad de ofrecer respuestas que no estén prefijadas, dando la libertad al usuario de seguir distintos caminos para alcanzar la solución, de usar funciones adicionales como usar una calculadora o buscar el código de un aeropuerto determinado, acciones no necesarias directamente en la resolución del ejercicio pero que pueden resultar interesantes para resolver una determinada situación.

Puesto que se ha trabajado para crearlo como un simulador web se ha seguido un esquema simple con varios pasos:



**Ilustración 11. Esquema del flujo de datos en el simulador.**

En el esquema de arriba se ven claramente los pasos que sigue la aplicación. El usuario interactúa con el terminal, la entrada se envía al respondedor que la procesa y carga las correspondientes clases Java, que comprueban si la entrada es correcta para el ejercicio indicado, si es necesario usando Funciones añadidas (ya veremos más adelante cómo funcionan) y en caso de coincidir con lo esperado se pasa al siguiente paso del ejercicio, devolviendo al usuario la respuesta esperada desde el terminal.

La idea es relativamente sencilla, además una aplicación web de estas características permite separar completamente la interfaz del núcleo de la aplicación, lo cual es una gran ventaja en muchos aspectos. Se ha trabajado para hacer la aplicación muy reutilizable y fácilmente ampliable, de ahí la posibilidad de agregar funciones añadidas, lo que unido a la posibilidad de cambiar la interfaz manteniendo la misma funcionalidad ofrece una gran cantidad de posibilidades de desarrollos futuros.

Esto no sería posible a menos que los ejercicios fuesen igualmente modificables y sencillos de crear. Para ello hemos trabajado para ofrecer un tipo de ejercicio que pueda servir realmente a cualquier propósito, aunque limitado como es lógico por las necesidades de este proyecto.

Cada uno está en un archivo XML, y este lenguaje nos permite establecer perfectamente los diferentes pasos del ejercicio, añadir etiquetas internas, información adicional y un sin fin de características según nuestras necesidades.

En los apartados posteriores veremos cómo funciona la aplicación y cómo se trabaja con los ejercicios de forma más específica.

### 3.5.1. Interfaz de la aplicación web- lado cliente

#### ► USO DE HTML / CSS

Para simular un terminal se ha utilizado HTML, CSS y Javascript. La estructura está creada en HTML, tomando de base un formulario y un área de texto para ofrecer la funcionalidad básica. El estilo es bastante simple, una pantalla de terminal es sobria por definición, así que lo que se ha hecho es ofrecer el mismo estilo.

Javascript es necesario para limitar los márgenes de la ventana del terminal y para ofrecer la funcionalidad de la aplicación, capturando la entrada del usuario y estableciendo el foco donde se necesita. Lo vemos más adelante.

#### ► MAQUETACIÓN TERMINAL AMADEUS

Sólo se han usado dos colores, azul para el fondo y blanco para el texto. Se ha dividido el contenido en dos partes, una para las respuestas del sistema y otra para introducir los códigos.

```

Queremos ver la disponibilidad para un vuelo entre Madrid y Pisa el 26
de Mayo. Luego sigue las instrucciones que aparezcan por pantalla.

USUARIODEPRUEBA
Introduzca su password:

ACCESO - USUARIO ACTIVO: 5

HE

Busca la disponibilidad entre Madrid y Pisa para el 26 de Mayo. Usa DAN
para buscar los códigos IATA.

AN26MAYMADPSA

** AMADEUS AVAILABILITY - AN **                75 MO 26MAY 0900
1  IB8223 C9 M9 T9 K9 V9 L9 Q9 MAD 4 PSA 1 0825 1030 0/320      02:05
2  AF1564 C9 M9 T9 K9 V9          MAD 4 PSA 2 0900 1115 0/737      02:15
3  LH7092 C9 M9 T9 K9 V9 L9 Q9 MAD 2 PSA 1 0945 1150 0/310      02:05

Cambia el destino a Paris Charles de Gaulle

>

```

Ilustración 12. Terminal del simulador

El resultado es exactamente el que esperábamos, y la impresión visual se apoya en la funcionalidad que ofrece.

## ► JAVASCRIPT Y AJAX

Como ya hemos dicho hemos utilizado Javascript para dar el aspecto de terminal. Las llamadas a los diferentes ejercicios establecen una ventana de unas dimensiones específicas y unas características fijas adicionales, como no mostrar la barra de direcciones, la barra de estados, etc.

Adicionalmente se ha usado Javascript dentro de la pantalla del terminal para fijar el foco del navegador en el campo para introducir el texto, en el proceso que recoge el código introducido y lo envía como una petición usando AJAX. También es importante el uso de Javascript en la recepción de la respuesta, la cual se recoge y se procesan sus diferentes partes (el nombre de usuario, el paso del ejercicio y el texto de la respuesta van incluidos en el mismo elemento). Se usan expresiones regulares y divisiones por “tokens” para extraer cada elemento y continuar con el ejercicio.

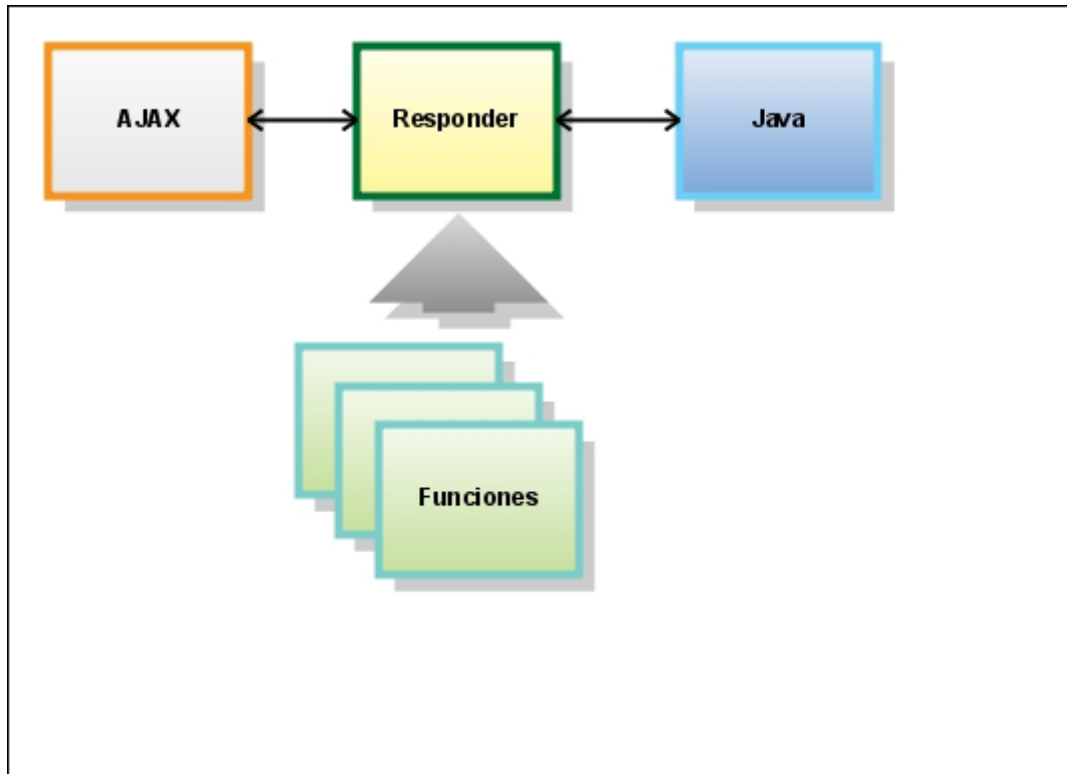
Ciertamente se podrían haber usado los valores de las sesiones que ofrece Tomcat y JSP para almacenar estos valores y no necesitar realizar el proceso descrito un poco más arriba, pero el proyecto a evolucionado desde la idea primordial, como es habitual, y son aspectos que no se han planteado de esa manera.

En particular esta funcionalidad estaba prevista, lo que ocurre es que la división de la interfaz en terminal y respondedor trajo unas consecuencias que facilitan este aspecto pero por falta de tiempo se ha quedado por el camino. Luego lo veremos más a fondo, pero se propone una interfaz de salida que se carga en la clase Java de entrada, la cual se encarga de recoger los diferentes valores de las variables que se consideren necesarias, y en este caso la página responder.jsp (el respondedor) se encargaría de recogerlas y trabajar con ellas, bien manteniéndolas como variables de sesión o de cualquier otra manera.

En este caso esto quedará para las líneas futuras y cambios propuestos.

Respecto al uso de AJAX, en este proyecto, si pretendíamos ofrecer una aplicación web con el funcionamiento típico de un terminal o de una consola, el uso de esta tecnología parecía especialmente adecuado.

A partir de este punto la interfaz se divide en dos partes, una es el terminal, del cual ya hemos hablado, y es el que se encarga de realizar la petición AJAX al respondedor. Luego procesa la respuesta como ya hemos explicado antes y vuelve a quedar a la espera de la siguiente acción del usuario.



**Ilustración 13. Carga de funciones desde JSP (responder.jsp)**

El respondedor es una página JSP que devuelve, en formato de texto plano, la respuesta de la aplicación realizada en Java además de una serie de variables añadidas que luego el terminal se encarga de procesar y usar para continuar con el funcionamiento de la aplicación.

Cuando el respondedor recibe un código éste viene acompañado del usuario que lo envía y del paso actual del ejercicio en el que se encuentra. El propio respondedor se encarga de cargar las Funciones añadidas a la aplicación, luego realiza una llamada a una interfaz de entrada y la aplicación Java devuelve una salida, que es la que posteriormente recibe el terminal.

### **3.5.2. JSP/ JAVA- lado servidor**

La aplicación funciona sobre Java y JSP, cargado desde Tomcat. Las páginas JSP se encargan de recibir la entrada y de llamar a las clases Java para procesarla, tal y como hemos explicado arriba.

Veamos ahora cómo trabaja la parte escrita en Java, que es donde se encuentra el motor de la aplicación y el apartado técnico más interesante.

## ► FUNCIONAMIENTO GENERAL

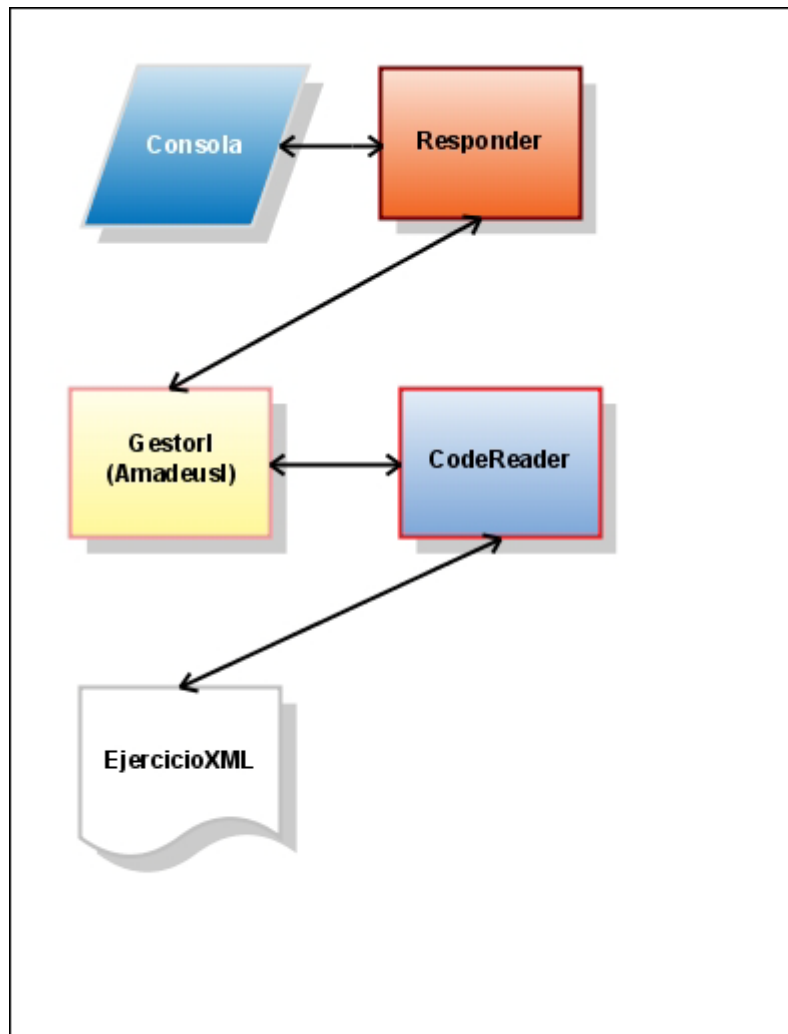
Como ya hemos explicado antes, el código introducido por el usuario llega al respondedor, una página JSP (responder.jsp) que además recibe el nombre del usuario y el estado del ejercicio.

Por defecto el nombre estará vacío y el estado será -3, indicando que es necesario mostrar el enunciado del ejercicio y prepararse para hacer login. Los estados indican lo siguiente:

- **-3** = Se debe mostrar el enunciado del ejercicio y avanzar al estado -2.
- **-2** = El usuario tiene que introducir un nombre, si es válido se avanza al estado -1
- **-1** = El usuario debe introducir la contraseña. Si es válida se avanza al estado 0 para iniciar el ejercicio.

Siguiendo estos pasos desde el terminal se introducirán los datos y el sistema responderá en función al estado en el que se encuentre.

Se han creado dos interfaces para tratar la entrada y la salida. Ambas son llamadas desde el respondedor, y desde ellas se interactúa con el resto de la aplicación. La interfaz de entrada recibe el nombre de GestorI y la de salida GestorO. La interfaz de entrada es implementada por la clase AmadeusI, que se encarga de recoger los datos del respondedor para luego llamar a la clase CodeReader. Esta clase es la principal, y su cometido es, dado un código, usuario y estado, resolver los distintos pasos permitidos para el ejercicio y comprobar si alguno se adecúa a lo introducido por el usuario.



**Ilustración 14. Flujo de I/O del simulador**

Cada estado tiene una serie de atributos que indican qué parámetros se requieren para acceder a él, su respuesta cuando se alcanza dicho estado o los que quedan accesibles desde él mismo, entre otros.

El grupo de estados del ejercicio se carga en una estructura tipo árbol, la cual facilita la navegación entre los diferentes elementos.

Desde el respondedor se cargan también las funciones añadidas que luego pueden ser referidas en los ejercicios. En las páginas siguientes vamos a ver cómo están organizadas las clases, su diagrama, diferentes aspectos de la aplicación que consideramos más interesantes o que deben ser explicados por su complejidad y el uso y funcionamiento de los ejercicios en XML.

## ► DIAGRAMA DE CLASES

Empecemos con el diagrama de clases. En primer lugar un esquema de los diferentes paquetes de la aplicación:

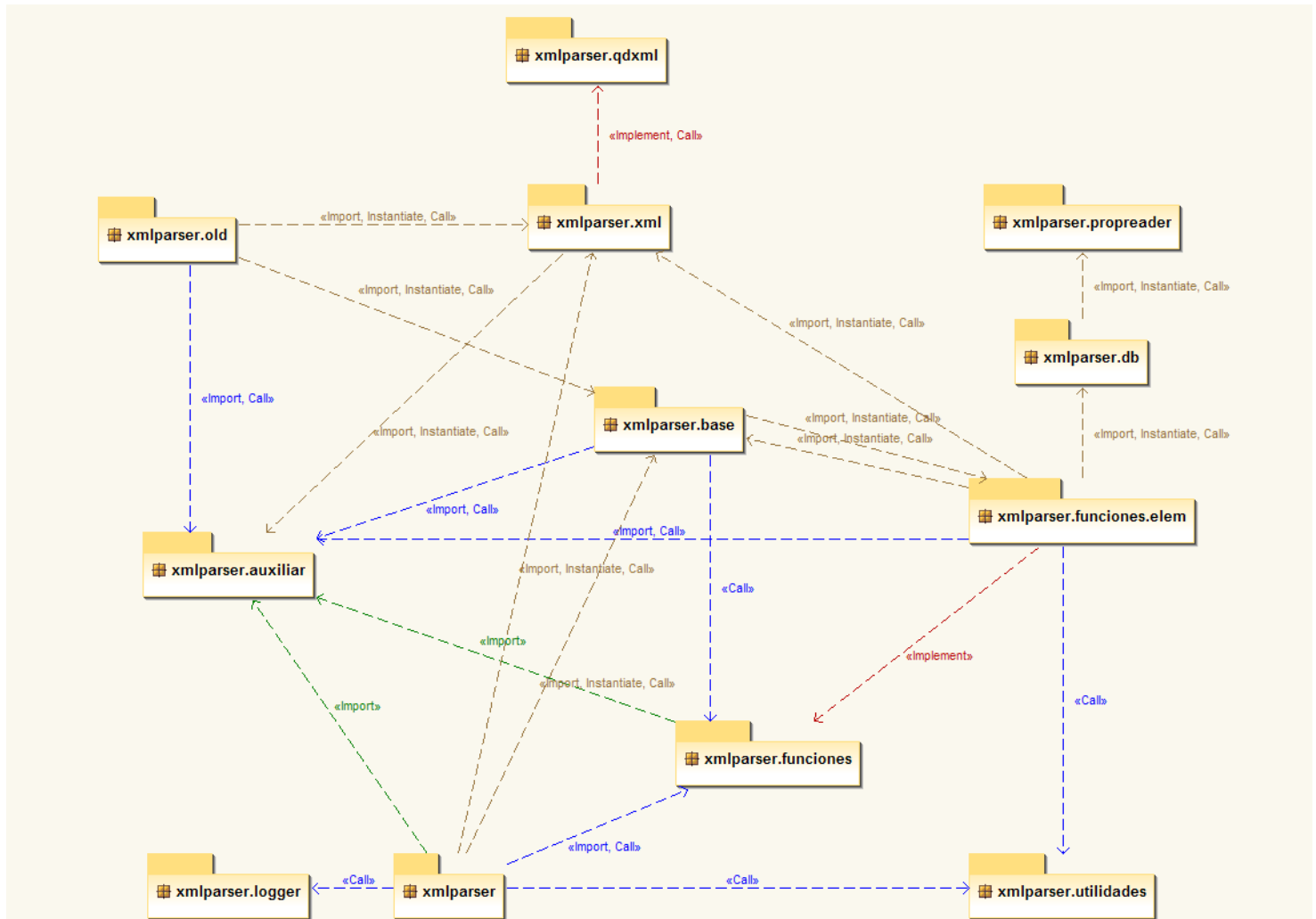


Ilustración 15. Diagrama de paquetes Java

con el que se puede observar las diferentes relaciones entre las partes del programa. El paquete base es donde se encuentra el motor que mueve a todos los demás. Como vemos hay muchos elementos que necesitan explicación.





A continuación vamos a comentar brevemente el cometido de cada clase englobadas en sus paquetes, para hacernos una idea general del funcionamiento de la aplicación:

GestorI: interfaz para manejar los datos de entrada de la aplicación.

GestorO: interfaz para manejar los datos de salida.

AmadeusI: implementación de GestorI. Carga CodeReader y recibe llamadas para cargar funciones externas.

AmadeusO: implementación de GestorO. Prepara las variables que devuelve la aplicación a una entrada determinada para que puedan ser recogidas.

### ***base***

CodeReader: es la clase principal. Se encarga de leer el archivo XML del ejercicio y en función del paso actual, del código introducido y del usuario devuelve una salida. La salida viene establecida en el propio archivo XML, bien como texto incluido o como llamada a una función externa.

NavArbol: contiene funciones para realizar búsquedas dentro del documento XML, dado que éste se carga en una estructura tipo árbol.

### ***auxiliar***

Arbol: es la clase que define la estructura tipo árbol.

Nodo: nodo de la estructura, contiene un objeto indeterminado además de enlazar con los nodos adyacentes (hijo y hermano).

Elemento: elemento enlazado desde cada nodo. Contiene los datos de una etiqueta del archivo xml (nombre, contenido, atributos).

### ***db***

Conector: da conexión con una base de datos MySQL. Los datos de conexión se encuentran en un archivo externo (mysql\_connect.ini).

### ***funciones***

Funciones: clase principal para el uso de funciones. Permite cargar funciones externas siempre que implementen una interfaz común (CargaFunciones) asignándoles un nombre, que luego podrán ser llamadas usando dicho nombre.

CargaFunciones: interfaz que debe ser implementada por las funciones para poder ser incluidas en la aplicación.

CargaSignOut: interfaz para la función de SignOut. Al tener una funcionalidad diferente de las otras requiere una aproximación distinta.

### ***funciones.elem***

FuncionSignInNameMoodle: función para hacer login usando como nombre de usuario uno incluido en el curso de Moodle. Los datos de conexión (nombre del curso, nombre corto, etc.) se encuentran en el archivo mysql\_connect.ini.

FuncionSignInPSWMoodle: igual que la anterior pero para la contraseña.

FuncionSignOut: función que realiza el logout. Usa los datos de conexión con Moodle para añadir una nota (apto o no apto).

FuncionDAN: función que recibe el nombre de una ciudad o aeropuerto y devuelve el código IATA. Los datos que utiliza esta función se encuentran en un archivo IATA.xml.

FuncionDAC: igual que la anterior para decodificar.

FuncionCalculadora: función que añade la posibilidad de usar acciones básicas de una calculadora de estilo Amadeus.

### ***logger***

Logger: esta clase se encarga de realizar logs de las acciones de los usuarios. Cada código introducido es registrado en la carpeta UserLogs.

### ***propreader***

PropertiesReader: clase de apoyo para leer archivos .ini o con estructura

similar.

### ***qdxml***

DocHandler: interfaz para trabajar con QDParser.

QDParser: versión modificada y simplificada del parser SAX de XML.

### ***utilidades***

Utilidades: clase con funcionalidades generales, como la modificación de cadenas de texto o la obtención de funciones resumen MD5 (necesaria para las contraseñas en la base de datos de moodle).

### ***xml***

MiParser: implementación de DocHandler para recoger un archivo XML e incluirlo en una estructura tipo árbol.

Con esto conocemos el contenido que hay en los paquetes y conocemos por encima qué función tiene cada una de las clases. Vamos a ver ahora el funcionamiento de las más importantes o interesantes.

## **► FUNCIONAMIENTO ESPECÍFICO**

Vamos a explicar detalladamente cómo funcionan algunas de las clases de la aplicación, las que consideramos más importantes o interesantes para el funcionamiento del programa.

En primer lugar hablaremos de CodeReader. Esta es la clase central, que se encarga de procesar el código, comprueba el archivo XML del ejercicio, llama a las funciones si fuera necesario y devuelve un texto que luego aparecerá por la consola.

La función que recibe la primera petición es:

```
public void processCode(String codigo, Arbol a, String estado, String path,  
String file, String usuario);
```

Lo primero que hace es comprobar el estado que recibe, si es -3 mostrará el

enunciado, si es -2 esperará un nombre de login y si es -1 esperará el password correspondiente. Para estos tres casos se realiza una llamada a funciones específicas, en el caso de mostrar el enunciado buscará el contenido de la etiqueta *<inicio>* del archivo XML, incluido en este momento en el árbol que recibe por parámetro *processCode*. Para el nombre y la contraseña se llama a funciones externas que en este caso conectarán con la base de datos de Moodle y comprobarán si los datos introducidos son correctos.

Puede que el usuario haya introducido el código *HE* que indica que pide la ayuda asociada al paso actual (indicada por el elemento *<ayuda>*) o el código *JO*, que cerraría la sesión.

A partir de aquí vienen los casos generales en los que el usuario introduce un código y se realiza todo el proceso hasta devolver una salida. Por lo tanto, lo primero que hace es llamar a la función *obtenerPosibles* que es la que se encarga de buscar entre los pasos posibles del ejercicio XML partiendo de la situación actual. Para ello cada estado tiene un elemento *<post>* que indica cuáles son a los que se puede llegar a partir del actual. En la lista de posibles se incluyen los estados globales, que pueden ser accedidos desde cualquier otro en cualquier momento.

Tras este paso se llama a la función *comprobarPosibles*, que recoge la lista que dejó el anterior y comprueba si alguno de los estados disponibles encaja con el código introducido por el usuario. Para ello cada estado tiene un elemento *<comprobar>* que permite realizar una comprobación directa entre el contenido y el código, pudiendo dividir la cadena de texto en varias partes o argumentos diferenciados, para facilitar así tanto la comprobación como la elaboración del ejercicio.

Tras esta segunda criba, si el ejercicio está correctamente construido, debería quedar un único estado posible en la lista, o ninguno en caso de que el usuario se haya confundido. Si este es el caso se llamará a la función *FinalError*, que tomará el estado actual del ejercicio y mostrará su mensaje de error asociado, incluido en el elemento *<error>*. En caso de que se hubiese introducido un código correctamente se llamaría a *FinalProcess*, que actualizaría el estado actual al del siguiente paso, recogería de dicho estado el contenido del elemento *<correcto>* y lo mostraría por pantalla<sup>35</sup>.

---

<sup>35</sup> Este texto que se muestra por pantalla, para su correcta visualización, está incluido entre etiquetas *<pre></pre>* puesto que muchos elementos de Amadeus se mostrarían por un terminal con un formato específico, que lo mantenemos aquí usando las etiquetas de preformato. El resto de

*FinalProcess* se encarga de comprobar si el contenido a mostrar tiene una función asociada, para obtener de ella los datos necesarios y realizar cualquier otra tarea.

En general el proceso es sencillo, en el caso típico hay tres posibles códigos: correcto, incorrecto y de ayuda. En el caso de que sea correcto se discrimina entre los posibles y se compara lo introducido con lo esperado, para escoger finalmente cuál será el siguiente paso en el ejercicio.

Hablaremos ahora de las funciones. La idea principal era que las funciones se pudieran cargar de forma externa, para así permitir añadidos posteriores y mejoras complementarias al funcionamiento básico. Para ello cada función debe tener los siguientes métodos implementados (controlados a través de una *interfaz*):

```
public int getLengthVariable(String var);  
  
public boolean checkVariable(String var);  
  
public String processVariable(String codigo, String path);
```

Las dos primeras comprueban que el código introducido sea correcto. Vemos que en este caso se pretende ofrecer una funcionalidad similar a los que se dispone de forma normal, siendo equivalente al contenido de *<comprobar>*. La tercera función es la que devuelve el contenido correcto. Por un lado recibe el *código* introducido por el usuario, y por otro el *path* o *camino* que puede ser de utilidad para archivos adicionales de configuración.

Una vez están creadas las funciones, deben cargarse en la aplicación para poderse utilizar en los ejercicios. Para ello existe *GestorI*, una interfaz que controla los elementos de entrada a la aplicación, como son el código a procesar y las funciones cargadas. La clase *AmadeusI* implementa dicha interfaz y carga las funciones en la clase *Funciones*.

La clase *Funciones* es una clase de acceso estático, por lo que no requiere ser instanciada para funcionar (o es instanciada por defecto al ejecutar la aplicación, si se prefiere). Implementa una tabla Hash<sup>36</sup> para recoger las funciones añadidas y utilizarlas después. Para llamarlas utiliza las posibilidades de herencia y polimorfismo

---

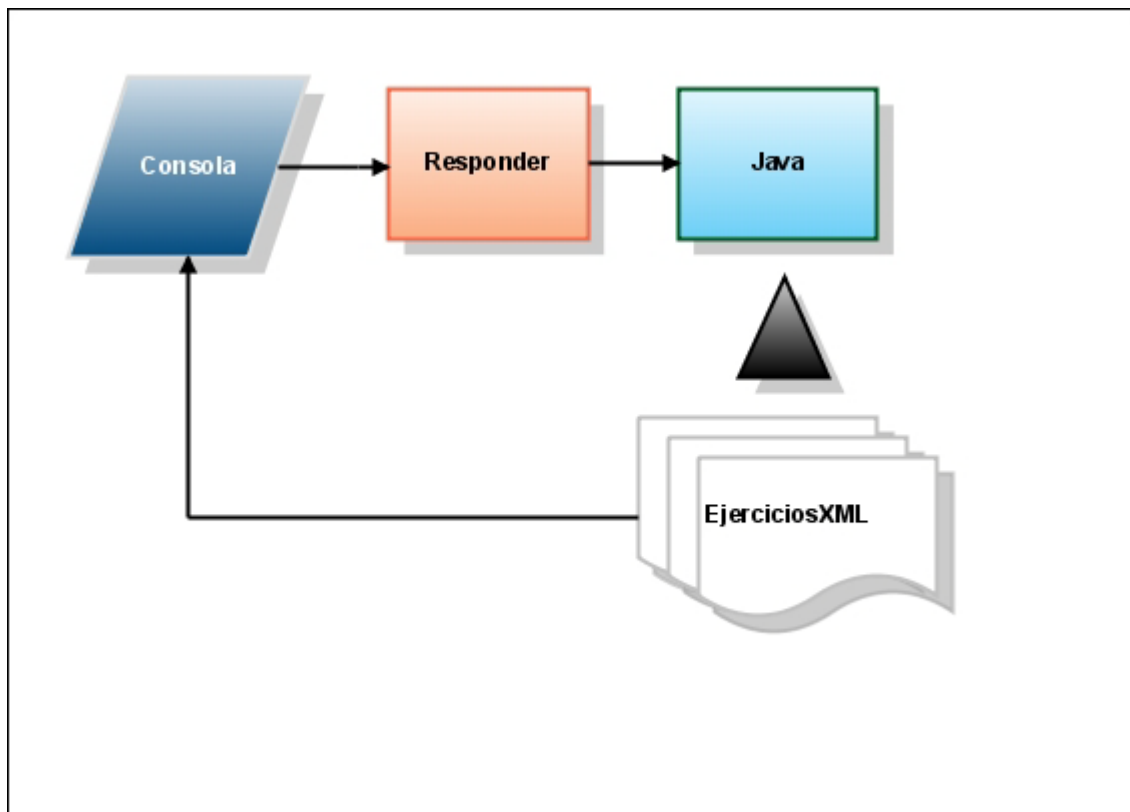
contenidos como los de *ayuda* o *error* se engloban entre las etiquetas *<p></p>* para definir párrafos individuales.

<sup>36</sup> *Hashtable* es una clase del paquete *util* de Java que permite asociar dos elementos en una tabla, de tal manera que cuando se pide el primero se obtiene el asociado. Es útil para glosarios de términos y definiciones o para el caso que nos ocupa, donde cada función tiene un nombre String asociado, y al realizar la petición devuelve el Objeto de la función.

que ofrece Java, ofreciendo los mismos métodos que están incluidos en la interfaz *CargaFunciones* y que por consiguiente estarán incluidos en todas las funciones, con la salvedad de que en este caso *Funciones* pide un nombre de función asociado para relacionar el Objeto con ese nombre.

### ► JSP, SESIONES Y CONTROL DE FLUJO

Siguiendo el flujo del programa, como hemos visto, quedan por explicar cómo funcionan los ejercicios XML, y su estructura en pasos o estados.



**Ilustración 17. Carga de ejercicios.**

La *consola* actualiza su contenido usando la técnica AJAX. Tras cada petición el *responder* devuelve un texto, que es lo que recibe de nuevo la *consola*. Para la obtención del texto *responder* carga los datos en la interfaz *GestorI* (*AmadeusI* en este caso) que es la que estableciendo el estado y el código de entrada se lo envía a *CodeReader*. Al mismo tiempo, *AmadeusI*, carga las funciones y carga el archivo XML en la estructura de árbol. En este árbol se encuentra el ejercicio, que es leído por *CodeReader* dependiendo del estado actual y del código introducido. Dependiendo de si es correcto o incorrecto se devuelve un contenido u otro.

Responder recoge de *AmadeusI* la respuesta dada por *CodeReader*, le adjunta el

estado y otros parámetros y eso es lo que recibe consola y muestra por pantalla.

### ► XML PARA EJERCICIOS

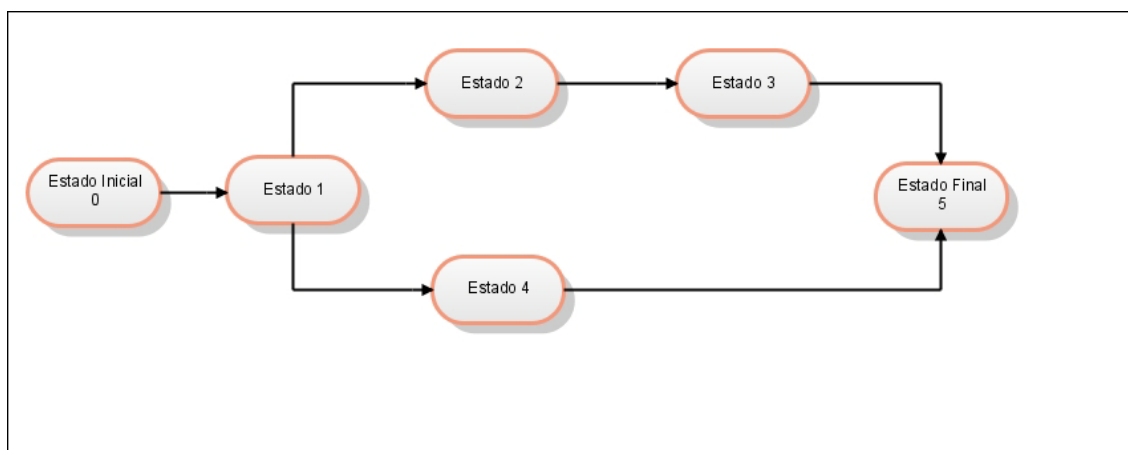
La simplicidad del XML y su facilidad de uso es lo que nos hizo decantarnos por este formato para los ejercicios. Además la posibilidad de dividir el contenido en etiquetas encajaba con nuestra idea de definir los ejercicios por pasos, ofreciendo diferentes caminos y opciones para llegar al resultado final.

Partimos de la idea de crear ejercicios completos que requiriesen una serie de pasos correlativos para llegar al final. La estructura era la siguiente:

**EJERCICIO = PASO1 → PASO2 → PASO3 → ... → PASOFINAL**

Sin embargo la versatilidad de Amadeus reside en que ofrece múltiples posibilidades para resolver una situación. Por ejemplo es posible que necesitemos introducir el nombre de dos pasajeros para realizar la reserva de un billete, lo cual podemos hacerlo poniendo primero uno y después el otro, lo que serían dos pasos, o haciéndolo con los dos a la vez, en un solo paso del ejercicio.

De este modo tenemos que contemplar también la posibilidad de ofrecer caminos alternativos:



**Ilustración 18. Diferentes caminos de un ejercicio tipo.**

Las etiquetas XML se definieron de la siguiente manera:



```

<estado id="6" codigo="AN">
  <post>
    <p>7</p>
  </post>
  <comprobar>
    <codigo>AN</codigo>
    <args>
      <a option="eq">14OCT</a>
      <a option="eq">CDG</a>
      <a option="eq">JFK</a>
      <a option="eq">1130</a>
      <a option="eq">/X</a>
      <a option="eq">LHR</a>
    </args>
  </comprobar>
  <correcto>
    ** AMADEUS AVAILABILITY - AN **           75  MO  14OCT  1130
    1  AF2548 C9 M9 T9 K9 V9           CDG 2 LHR 2 1200 1415   0/310
        BA5151 C9 M9 T9 K9 V9 L9 Q9 LHR 2 JFK 1 1535 1810+1 0/747   18:10

    ¡Enhorabuena! ya puedes salir del ejercicio.
  </correcto>
  <error>
    ¡Error! PISTA: para salir introduce JO
  </error>
  <ayuda>
    Ya puedes salir del ejercicio.
  </ayuda>
</estado>

```

**Ilustración 19. Imagen de un estado de un ejercicio XML**

Lo que nos permite, ahora sí, realizar ejercicios con diferentes caminos para llegar al final.

Como podemos observar hay diferentes apartados en cada estado, el apartado post indica los estados posteriores a los que se puede acceder desde este, el apartado correcto indica lo que se mostrará cuando se llegue a este paso, los apartados dentro de comprobar indican frente a qué se comprobará el código que introduzca el usuario si este estado se encuentra entre los que son accesibles desde el anterior. Además se han incluido etiquetas de ayuda y de error, que muestran el contenido al usuario si pide ayuda desde este estado o si no introduce correctamente el código siguiente respectivamente.

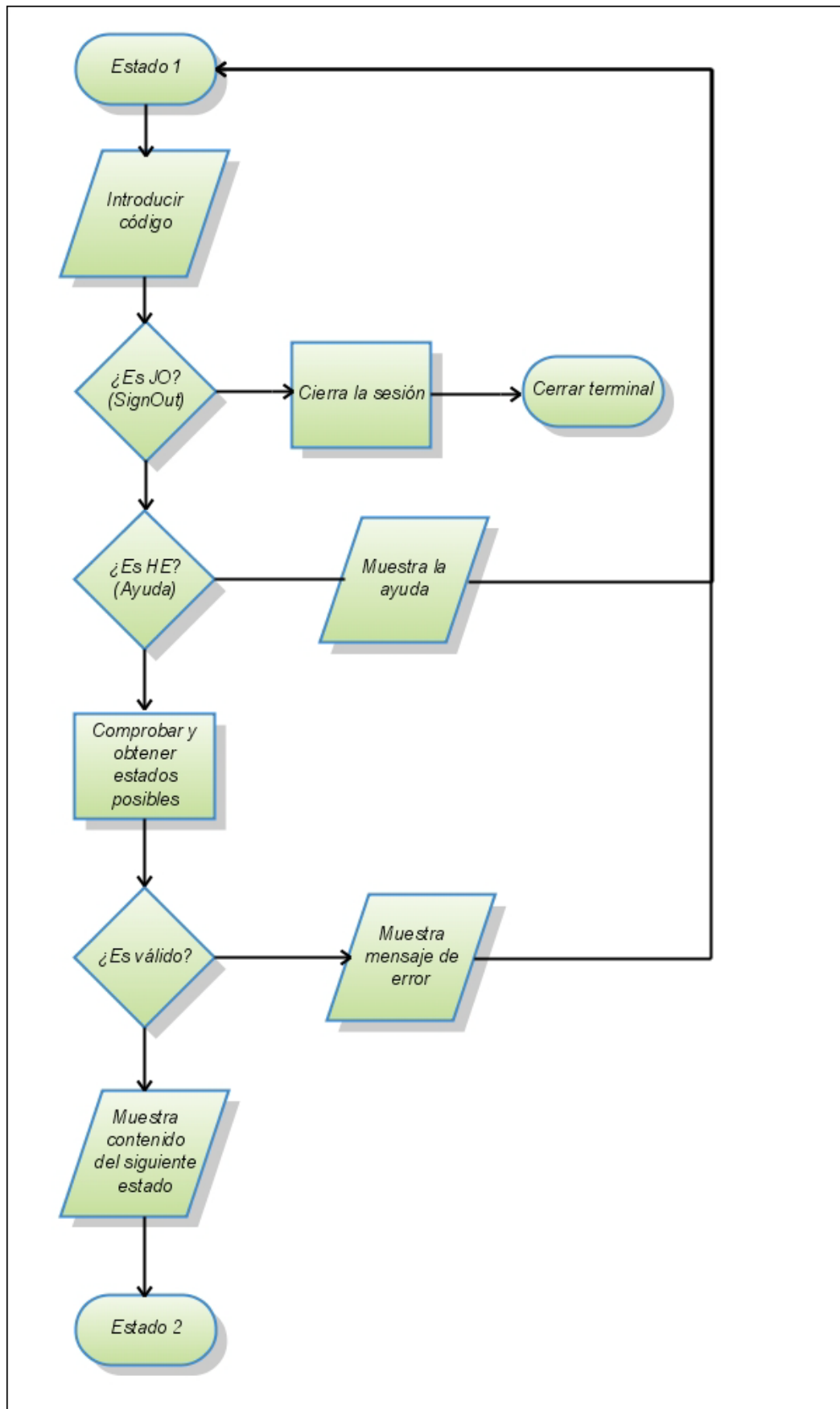


Ilustración 20. Esquema de flujo para acceder de un estado a otro

Sin embargo esto no era suficiente, puesto que existen códigos que deberían poder ser introducidos en cualquier momento ya que aunque no son indispensables para el ejercicio si muestran información que puede ser útil para resolverlo.

Como hemos visto los pasos de cada ejercicio están numerados para poder seguir un orden. Los estados a los que se pueda acceder globalmente tendrán una letra en lugar de un número. No modifican el estado actual, es decir, si se introduce el código para acceder a un estado global éste mostrará su contenido y el estado se mantendrá en el paso anterior. De esta manera tenemos todas las posibilidades que necesitamos para establecer el flujo del ejercicio.

Para terminar con la idea de los ejercicios en XML añadimos un atributo para aceptar funciones. Dicho atributo se usa cuando se comprueba si el código que se ha introducido corresponde con el de dicho estado y/o cuando se debe mostrar la información por haber accedido correctamente a este estado.

La función requerida en cada estado viene indicada mediante una palabra clave, que será la misma que se haya introducido al cargar la función en la interfaz de la aplicación. De este modo todas las funciones siguen el mismo proceso y ofrecen el resultado de manera adecuada.

Con este último paso hemos conseguido ampliar las posibilidades enormemente, puesto que no sólo contaremos con la capacidad de establecer ejercicios que permitan seguir distintos caminos para finalizarlos, si no que además se ofrecen herramientas globales útiles para el usuario como son las funciones de codificación de aeropuertos.

De este modo se garantiza un funcionamiento atractivo para más adelante, si se cree necesario, poder desarrollar funcionalidades añadidas.

Para un futuro se propondrá la creación de una aplicación que ofrezca una interfaz gráfica para generar los ejercicios, lo cual, añadido a la creación de nuevas funciones puede llevar a realizar ejercicios mucho más detallados e interesantes.

#### ► **PARSER XML**

El parser de XML usado es una versión mínima del parser SAX, un parser basado en eventos pero con mucha menos funcionalidad. Puesto que no se requerían más que las capacidades básicas es una elección adecuada, ya que la intención principal era recoger el contenido de un archivo XML e introducirlo en una estructura de árbol

para trabajar fácilmente con él.

Esta versión reducida<sup>37</sup> contiene dos archivos java, uno es *QDParser.java*, que es donde se encuentra el código obtenido de SAX. Se han eliminado funcionalidades que no son necesarias pero mantiene lo esencial para el funcionamiento. El otro archivo es *DocHandler.java*, el cual es una interfaz para manejar *QDParser*, y está implementada por *MiParser.java*.

*MiParser* contiene una implementación sencilla, que simplemente va aprovechando los eventos que devuelve *QDParser* para incluir el contenido del XML en un árbol de etiquetas xml.

## ► OTROS DETALLES

Con esta aplicación se ha intentado en todo momento ofrecer la posibilidad de mejorar y adaptar su uso a las condiciones que se requieran. No sólo para añadir más ejercicios o más funciones, si no incluso pudiendo cambiar por completo la finalidad de la aplicación.

Si tomamos los JSP de consola y responder y los modificamos, cambiando el diseño, por ejemplo hacia un sistema visual de Amadeus, traduciendo la pulsación de botones en órdenes con formato de códigos transaccionales, y luego se cambian los ejercicios para ofrecer respuestas acordes con la nueva interfaz visual, tenemos una aplicación diferente pero que parte de la misma base que la anterior.

No es necesario quedarse sólo con una mejora visual de la interfaz de Amadeus, es posible crear una interfaz pensada para una finalidad completamente diferente y establecer las diferentes respuestas del sistema en los archivos XML. Está claro que la funcionalidad siempre va a basarse en pasos secuenciales como ya hemos explicado en los puntos anteriores, pero a pesar de dichas limitaciones puede resultar interesante.

---

<sup>37</sup> Basada en la ofrecida en la web de Javaworld, desicada al mundo de la programación en Java. En ella se encuentra más información sobre el parser. Disponible en: <http://www.javaworld.com/javatips/jw-javatip128.html> [Consulta 5-05-09]

### 3.5.3. Conexión con ejercicios moodle.

La conexión de los ejercicios con Moodle se realiza en dos momentos: el SignIn y el SignOut. Para ello, usa dos funciones externas que realizan una conexión con la BD en la que se encuentra almacenada la información de Moodle.

Para el SignIn se comprueban el nombre y la contraseña en la base de datos MySQL de Moodle y si es correcto se sigue adelante. De esta manera el usuario es establecido como el actual en la aplicación. No tiene mayor interés hasta que se hace SignOut.

La función de SignOut tiene dos partes, desconectar el sistema borrando el usuario activo volviendo a poner el estado del ejercicio en el inicial, y poner la nota adecuada.

El estado final tendrá el atributo “final=true”, lo que indica que si al realizar SignOut el usuario se encuentra en dicho estado la nota será Apto. En caso contrario será NoApto.

Hay que recordar que las funciones de SignIn y SignOut pueden modificarse o cambiarse por otras siempre que respeten la interfaz para cargar estas funciones. Esto quiere decir que si se deben buscar los usuarios en una base de datos que no sea la de Moodle, o las notas a poner son diferentes, por ejemplo, se puede realizar cambiando las funciones encargadas de hacerlo.

#### ► ACTIVIDADES OFFLINE VALORADAS

Como ya explicamos en su momento, para unir Moodle con los ejercicios en Tomcat lo hacemos usando el módulo de *Actividad Offline*. Este módulo es adecuado porque permite colocar un enlace a la página que se encargue de lanzar los ejercicios de cada tema, al mismo tiempo que nos da un espacio para incluir las notas de los ejercicios, cosa que se hace automáticamente como hemos visto en el punto anterior.

Queda propuesto para un futuro el desarrollo de un módulo que permita realizar la misma función, pero específico para los ejercicios del simulador de Amadeus.

#### ► TABLAS MYSQL RELACIONADAS

Las tablas y campos relacionados con la consulta de usuarios o la inserción de notas están escritos en el archivo *mysql\_connect.ini* para facilitar su modificación.

Estas variables son las que se cargan en la función de SignIn o SignOut para usarlas en los datos de conexión. Principalmente se deberán cambiar tres variables, casi con total seguridad, dependiendo de dónde se instale: la dirección IP para conectarse a la base de datos, el nombre de usuario con acceso a la BD de Moodle y el password.

Si se crearan funciones de login y logout diferentes se podrían incluir en ese mismo archivo las variables que fuesen necesarias y usar las clases *PropertiesReader* y *Conector* para acceder a ellas.

### ► VALORES MODIFICABLES

Específicamente las notas de los ejercicios Offline de Moodle se encuentran en la tabla *mdl\_assignment\_submissions*<sup>38</sup>, que contiene las notas puestas a los ejercicios y a la cual se accede, comprobando qué ejercicio es el asignado, desde la tabla *mdl\_assignment*, que sería el ejercicio perteneciente al curso relacionado, que se encuentra en la tabla *mdl\_course*. Así relacionando al alumno que se encuentra en *mdl\_user*, con la tabla que le identifica como perteneciente al curso que es *mdl\_course\_display*, se puede realizar el seguimiento y la aplicación de notas de manera externa a Moodle sin modificar el funcionamiento ni introducir errores.

El aspecto final con dos usuarios de ejemplo que han realizado dos ejercicios de calificación cada uno sería:

Curso de AMADEUS				
Nombre / Apellidos ↑	Simulador ↓	Simulador 2 ↓	Simulador ↓	̄ Total del curso ↓
😊 Mellon Ganyan III	100,00	100,00	-	100,00
😊 Manolo Hu	100,00	0,00	-	50,00
Promedio general	100,00	50,00	Respuesta No Apto	75,00

**Ilustración 21. Calificaciones en Moodle**

De este modo, desde el panel de control de calificaciones dentro de Moodle, puede el profesor acceder a las notas de sus alumnos y ver los resultados tanto de los ejercicios de cuestionario como de los de simulación de Amadeus.

<sup>38</sup> En este ejemplo las tablas tienen asignado el prefijo *mdl\_* pero no es específicamente obligatorio, en la instalación de Moodle se puede modificar, por lo que el nombre de las tablas puede variar a causa de esto.

## 4. EXPERIMENTACIÓN/ PRUEBA CONCEPTO

En este momento ya tenemos el curso creado.

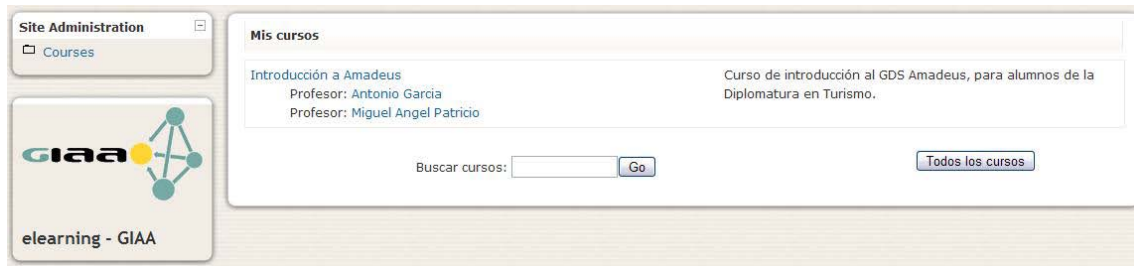


Ilustración 22. Pantalla de [www.giaa.inf.uc3m.es/moodle](http://www.giaa.inf.uc3m.es/moodle)

Con todos sus contenidos en orden:



Ilustración 23. Imagen de un tema de teoría.

Vamos a ver cómo usarlo accediendo tanto siendo estudiante como profesor.

## **4.1. Curso**

La web para acceder al curso es la de Moodle del Grupo de Inteligencia Artificial Aplicada: [www.giaa.inf.uc3m.es/moodle](http://www.giaa.inf.uc3m.es/moodle)

Bajo el nombre de “Introducción a Amadeus” se encuentran los temas y ejercicios que hemos desarrollado durante el proyecto.

Se debe asignar un profesor de la asignatura e introducir a los alumnos para comenzar el curso. Por defecto el profesor deberá ocultar los temas que no hayan sido estudiados, e ir haciéndolos visibles según vaya avanzando el cuatrimestre. Otra posibilidad sería establecer que se abra un nuevo tema por semana. De cualquier manera queda en las manos del profesor.

### **► PRUEBA DE ALUMNOS EN EL SIMULADOR**

Desde la perspectiva de un alumno veremos solamente las clases que el profesor haya puesto visibles.



1


## Presentación

En esta lección nos familiarizaremos con los elementos del curso e introduciremos los temas que vamos a ver durante el mismo.

Los puntos a tratar son:

- Bienvenida
- Los códigos de transacción
- Los ejercicios

---

 [Teoría Tema 1](#)

2


## Introducción a AMADEUS


Primer vistazo al sistema AMADEUS y toma de contacto con los ejercicios. Ejercicios básicos en el simulador.


Los puntos a tratar son:

- Los Sistemas de Distribución Global
- El sistema Amadeus
- El terminal
- Sign In
- Función calculadora

---

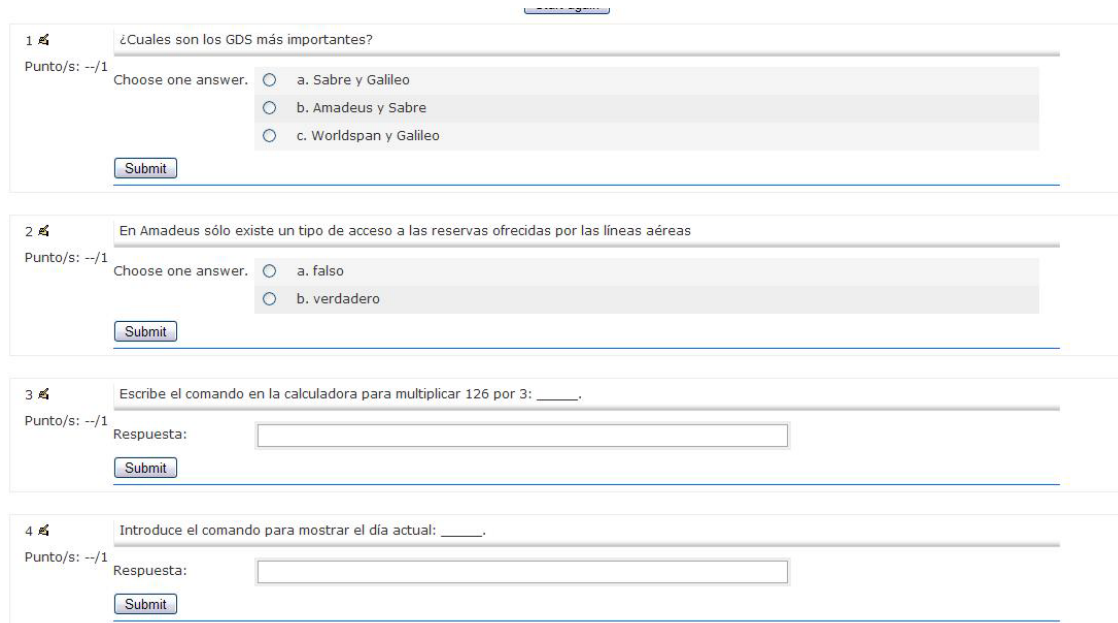
 [Teoría Tema 2](#)

 [Preguntas Tema 2](#)

 [Ejercicios simulador tema 2](#)

**Ilustración 24. Los primeros temas del curso.**

Se puede acceder tanto a la teoría como a los ejercicios,



The screenshot shows a Moodle quiz interface with four questions. Each question has a 'Punto/s: --/1' label and a 'Submit' button.

- Question 1: ¿Cuales son los GDS más importantes?  
Choose one answer.  
☐ a. Sabre y Galileo  
☐ b. Amadeus y Sabre  
☐ c. Worldspan y Galileo
- Question 2: En Amadeus sólo existe un tipo de acceso a las reservas ofrecidas por las líneas aéreas  
Choose one answer.  
☐ a. falso  
☐ b. verdadero
- Question 3: Escribe el comando en la calculadora para multiplicar 126 por 3: \_\_\_\_\_.  
Respuesta:
- Question 4: Introduce el comando para mostrar el día actual: \_\_\_\_\_.  
Respuesta:

**Ilustración 25. Preguntas del cuestionario.**

además de, por supuesto, a los ejercicios del simulador

```

Después de hacer Sign In prueba a codificar y decodificar aeropuertos.
Este ejercicio consiste en familiarizarse con los códigos DAN y DAC. Se
propone buscar las siguientes ciudades: Tbilisi, Prague, Cleburne,
Mokuleia, Istanbul y Agadez; y buscar los siguientes códigos IATA: CDG,
ERF, CMN, NEV, VGT, JFK y SVQ. Por supuesto puedes buscar otros códigos
o ciudades, teniendo en cuenta que el nombre suele encontrarse en
inglés.

USUARIODEPRUEBA
Introduzca su password:
ACCESO - USUARIO ACTIVO: 5
HE

Puedes usar la calculadora y codificar/decodificar aeropuertos. Los
propuestos son Tbilisi, Prague, Cleburne, Mokuleia, Istanbul, Agadez y
CDG, ERF, CMN, NEV, VGT, JFK, SVQ. A partir de este ejercicio tendrás
siempre disponibles la función calculadora y los códigos DAN y DAC para
los aeropuertos. Cuando hayas probado suficientes puedes salir del
simulador.

> |

```

**Ilustración 26. Ejercicio en el terminal del simulador.**

La interfaz de Moodle es muy intuitiva y con unos pocos clics de ratón el alumno tiene acceso a toda la teoría y práctica que queramos ofrecer.

## 5. CONCLUSIONES

Como finalización a la memoria del Proyecto de Fin de Carrera vamos a presentar las conclusiones de nuestro trabajo y las líneas futuras propuestas para su continuación.

### 5.1. Conclusiones

1. Partiendo desde el objetivo general de ofrecer una plataforma un curso de Introducción a Amadeus, el curso está funcionando en el servidor, incluyendo el simulador, preparado para recibir a los alumnos y para integrarse en las clases de la asignatura.

2. Otro de los objetivos que fue tomando más fuerza mientras avanzaba el proyecto fue la necesidad de ampliación del simulador. Partiendo del objetivo de utilizar un simulador sencillo, nos dimos cuenta a lo largo de la investigación de que era necesario permitir nuevas funciones. Una aplicación que tomase ejercicios por pasos y fuese avanzando sin más añadidos no tenía sentido, puesto que, salvando la presentación como un terminal, es algo que se podía hacer sin necesidad de crear un nuevo programa, incluso usando simplemente Moodle. De aquí sale la necesidad de permitir nuevas funciones, y creemos que con el sistema actual hemos cumplido también este objetivo.

3. Con respecto a la reutilización del código para realizar otro tipo de ejercicios, creemos que con la implementación actual es perfectamente posible adaptar la interfaz y ofrecer otro tipo de contenidos y ejercicios aprovechando las capacidades de la aplicación actual.

4. Como en todo proyecto, existe un límite que debe establecerse para no excederse en el tiempo de desarrollo ni en la magnitud del proyecto. Para ello se definen unas líneas futuras que indican las modificaciones que serían necesarias para seguir avanzando en el proyecto y convertir la idea original en una realidad más completa.

## 5.2. Líneas futuras

Como *líneas futuras* queda propuesto lo siguiente:

1. Incluir los contenidos del curso en lecciones. Moodle no ofrece directamente la posibilidad de limitar el acceso a contenidos del curso en función a la valoración del estudiante, pero sí tiene esta posibilidad a través de las *Lecciones*. De esta manera nos aseguraríamos que el alumno ha conseguido una calificación mínima para poder acceder al tema siguiente, haciendo menos necesaria la intervención directa del profesor y dejándole libertad para actuar cuando sea necesario.
2. La creación de un módulo dentro de Moodle que permitiera añadir ejercicios del simulador de una forma más directa. De esta forma se podría escoger directamente del desplegable de actividades una que fuera *Simulador* y a partir de ahí indicar los archivos de ejercicios incluidos.
3. Sería deseable la creación de una aplicación que ayudase de forma visual a la creación de ejercicios en XML. Actualmente el proceso es sencillo pero puede suponer una barrera importante para aquéllos que no estén acostumbrados al XML.
4. Actualmente no se comprueba que el XML esté bien formado y que el contenido sea consistente con lo que la aplicación esperará encontrar. Es algo que se puede solucionar fácilmente de forma conjunta a la propuesta del párrafo anterior.
5. La interfaz *GestorO*, implementada por *AmadeusO*, no se usa en la versión actual. La idea de esta parte del programa es tener una manera de acceder a los diferentes datos que tuviera *CodeReader* de una forma más simple e intuitiva. Actualmente se devuelven todos los datos (usuario, paso del ejercicio, respuesta) en una cadena de texto que debe ser procesada posteriormente.

### ► SIMULADOR COMPLETO

Hemos intentado dejar la máxima libertad para crear los ejercicios de la manera más variada posible y para aceptar el uso de funciones adicionales que mejorasen el comportamiento del simulador. Una de las premisas que tuvimos para comenzar el proyecto, y que planteamos en este documento, fue la necesidad de “*simular o crear ejercicios*”. Nos pareció más adecuada la creación de ejercicios porque era una solución más simple y directa a la necesidad que se presentaba. Partiendo de esto resultaría difícil extender la aplicación a la funcionalidad de un simulador completo, un

mini-CRS, a pesar de la libertad que ofrecen las *funciones* para mostrar contenido. Sin embargo creemos que se puede alcanzar un término medio con la adición de nuevos componentes, y que además facilitaría la creación de ejercicios al liberar la búsqueda de contenidos en cada paso y centrarse en las diferentes opciones para resolver una situación.

## APÉNDICES

### Manual del profesor

1. Qué debe saber un profesor sobre el curso
2. Cómo modificar aspectos de Moodle
3. Cómo ver y cambiar los GIFT directamente de Moodle
4. Cómo modificar ejercicios XML de forma sencilla
5. Cómo crear ejercicios
6. Cómo ve la calificación de los alumnos

### Manual técnico/ instalación

1. Requisitos
2. Enlaces para obtener el software
3. Propuesta AMP
4. Instalación y My SQL
5. Instalación Apache
6. Instalación PHP
7. Instalación de Tomcat
8. Instalación de Moodle
9. Instalación del Curso en Moodle
10. Instalación de la aplicación en Tomcat
11. Añadir funciones al simulador

## MANUAL DEL PROFESOR

### 1. Qué debe saber un profesor sobre el curso.

Este es un curso de *Introducción a Amadeus*, donde se tratan los conocimientos básicos necesarios para que, en diez lecciones sencillas, alguien que no esté familiarizado con el CRS Amadeus sea capaz de realizar una búsqueda de vuelo y una reserva para uno o varios pasajeros, teniendo en cuenta posibles incidencias y diferentes opciones a lo largo del proceso.

Pretende ser sencillo para el estudiante y dejar pocas dudas en los conceptos clave para facilitar la labor del profesor. De esta manera se ayuda al alumno a centrarse en los contenidos y, si se siente atraído por el temario, se le incita a profundizar más con ayuda del profesor.

El curso está dividido en diez temas, siendo el primero de introducción, los otros nueve tratan un aspecto determinado de Amadeus, como son las búsquedas, las reservas, las colas o los registros de pasajeros.

Cada tema después del primero tiene dos tipos de ejercicios evaluables: unos tipo test en los que, dependiendo de la pregunta, se requiere un tipo de respuesta u otra (puede ser múltiple, de rellenar el hueco, respuesta numérica, etc.); otros se realizan en un entorno de simulación de Amadeus, con la intención de permitir la resolución de situaciones más complejas teniendo la posibilidad de tomar diferentes decisiones y/o caminos para alcanzar el resultado final.

Como profesor se podrán modificar multitud de aspectos de moodle, incluyendo las lecciones o la forma de presentar los temas a los alumnos, los ejercicios tipo test y los del simulador.

## **2. Cómo modificar aspectos de Moodle.**

Usando las opciones que aparecen al activar la edición se puede modificar cualquier contenido de la página del curso, desde cambiar elementos de sitio o tabularlos, hasta modificar contenidos de teoría o ejercicios, incluso añadir o quitar paneles de consulta para los alumnos y/o profesores.

Al lado de cada elemento modificable aparecen unos iconos (flechas, cruces y otros símbolos) que indican la acción que se quiere realizar. Es una forma sencilla y útil de darle al curso un aspecto más atractivo, o de enfocar la atención en determinados puntos añadiendo notas o documentos que refuercen ideas.

En general Moodle ofrece un entorno y unas herramientas muy eficaces para facilitar el trabajo del educador y poder mostrar al alumno los contenidos de la forma más adecuada.

## **3. Cómo ver y cambiar los GIFT directamente dentro de Moodle**

Los ejercicios en formato GIFT se han importado desde un archivo de texto en el que estaban incluidos todos en conjunto. Si embargo Moodle tiene una manera más visual y directa de crear ejercicios y preguntas. En este caso es tan sencillo como modificar un cuestionario (editando la página, el icono de *actualizar*) o acceder desde la tabla de actividades al enlace de *Cuestionarios*, donde se debe escoger el que se desea modificar (se pueden añadir preguntas directamente).

Si se escoge un cuestionario se debe ir al final de la página (si se desea se puede cambiar cualquier opción del cuestionario) y allí seleccionar el botón de *Salvar* y

*mostrar* o *Save and view*<sup>39</sup>, Una vez ahí los pasos son los mismos tanto si se editó el cuestionario como si se accedió desde el menú de actividades: en la pestaña editar se muestran las preguntas del ejercicio seleccionado y la posibilidad de añadir nuevas preguntas de diversos tipos.

La manera habitual de crear un ejercicio es crear primero la pregunta, asignarla a una categoría (Moodle crea automáticamente categorías generales para cada tema) y después asignarla a un ejercicio. Hay un gran número de opciones, y aunque la interfaz es clara y comprensible requiere estar familiarizado con la plataforma y con la manera que tiene Moodle de tratar los ejercicios. Por ello recomendamos, si el profesor no ha usado Moodle antes, la creación de un curso de prueba donde practicar la creación de temas y actividades previamente a crearlas dentro del curso final.

#### **4. Cómo modificar ejercicios XML de forma sencilla.**

Los ejercicios se encuentran dentro de la carpeta llamada "EjerciciosXML" en el directorio SimAmadeus de las aplicaciones de Tomcat (es decir, carpeta /Tomcat/webapps/SimAmadeus/EjerciciosXML/). Se necesita acceso al servidor de Tomcat para modificar los ejercicios, no se puede hacer a través de Moodle puesto que, aunque los ejercicios de simulación complementan a los de Moodle, se encuentran en un servidor diferente y una aplicación distinta.

Los ejercicios tienen una estructura basada en *pasos* o *estados*, de manera que cambiar cualquier ejercicio consiste en modificar o añadir uno o más estados.

Un ejercicio para ser válido debe tener al menos dos estados, uno es el estado 0, que es sigue la misma base para todos los ejercicios, porque es desde el que se parte para comenzar el ejercicio. Un estado normal será:

```
<estado id="1" codigo="DNA">
  <post>
    <p>2</p>
    <p>3</p>
  </post>
  <comprobar>
    <codigo>DNA</codigo>
    <args>
      <a option="eq">Lufthansa</a>
```

---

<sup>39</sup> Hay que tener en cuenta que, pese a que Moodle dispone de traducciones e numerosos idiomas, es posible que el entorno en el que se encuentre instalado no tenga el idioma correspondiente o lo tenga de manera parcial y la interfaz estará en parte o totalmente en inglés, que es el idioma por defecto. Durante la instalación de Moodle es posible que pregunte si se desea instalar el paquete de idioma, lo cual solucionará la mayoría de estos problemas.



```

        </args>
    </comprobar>
    <correcto>
    LH/DLH LUFTHANSA

```

Ahora pide el código de Japón (en inglés Japan)

```

        </correcto>
        <error>
            ¡Error! PISTA: para determinar el país usa DC
        </error>
        <ayuda>
            Comprueba el código de país de Japón (Japan).
        </ayuda>
    </estado>

```

Vamos a explicar detenidamente en qué consiste cada parte. Se sigue un estilo XML de etiquetas para indicar cada elemento, deben abrirse, añadir atributos, poner contenido y cerrarse, siguiendo el esquema:

```

<etiqueta atributo1="valor1" atributo2="valor2">
    contenido
</etiqueta>

```

No es necesario el salto de línea entre las etiquetas y el contenido, y el número de atributos puede ser 0. Además, en la parte del contenido se pueden añadir etiquetas:

```

<etiqueta>
    <subetiqueta>contenido1</subetiqueta>
    <subetiqueta>contenido2</subetiqueta>
</etiqueta>

```

El cierre de las etiquetas se hace con el mismo nombre que la etiqueta de apertura precedido de la barra "/", como se ve en los ejemplos. Es importante cerrar las etiquetas que se abren o el ejercicio podría no ser leído.

Conociendo la manera de escribir etiquetas y contenidos en XML podemos describir los elementos que aparecen en un estado del ejercicio.

En primer lugar aparece la etiqueta:

```

<estado id="1" codigo="DNA">

```

donde el nombre de la etiqueta es estado, seguido del *id*, que es el número que indica el paso del ejercicio en el que nos encontramos, y además contiene el atributo *codigo*, que contiene el código que debe ser introducido. Este segundo atributo es opcional, pues no es comprobado por la aplicación, sólo sirve de referencia visual para crear los

ejercicios.

```
<post>
  <p>2</p>
  <p>3</p>
</post>
```

Aquí se ve la etiqueta *post*, que contiene etiquetas *p*, dentro de las cuales se especifican los estados a los que se puede ir después de éste. Cada estado siempre debe tener al menos un elemento *post* con un elemento *p* para continuar con el ejercicio, a menos que se trate del último.

```
<comprobar>
  <codigo>DNA</codigo>
  <args>
    <a option="eq">Lufthansa</a>
  </args>
</comprobar>
```

La etiqueta *comprobar* indica el texto que debe tener el código de entrada para ser aceptado y poder pasar a este estado. Tiene una subetiqueta *codigo*, la cual es opcional y se usa para especificar de una forma más visual la parte principal de la comprobación. También tiene etiquetas *args*, que permiten añadir más texto al código esperado. Requieren necesariamente el atributo *option* con el valor “eq” tal y como se muestra en el ejemplo.

```
<correcto>
LH/DLH LUFTHANSA

Ahora pide el código de Japón (en inglés Japan)
</correcto>
```

La siguiente etiqueta es *correcto*, que indica qué contenido se mostrará si se pasan las comprobaciones de la etiqueta anterior. Por defecto los contenidos de *correcto* se muestran “preformateados”, lo que quiere decir que en la página web aparecerán entre las etiquetas *<pre>**</pre>*. A efectos prácticos el contenido debería estar sin tabular para que se mostrase pegado al borde de la pantalla. La necesidad de trabajar con texto preformateado proviene de la naturaleza del terminal, con limitaciones de espacio en el largo de las líneas que obligan a prestar especial atención a la forma de presentar el texto.

```
<error>
¡Error! PISTA: para determinar el país usa DC
</error>
```

```
<ayuda>
    Comprueba el código de país de Japón (Japan).
</ayuda>
```

Los últimos dos bloques del estado hacen referencia a dos elementos de ayuda para superar el estado actual. El contenido de *error* se muestra cuando se introduce un código que no puede cambiar del estado actual a uno de los contenidos en *post*, mientras que el contenido de *ayuda* se muestra cuando el usuario introduce el código *HE* para pedir ayuda. Hay que tener en cuenta que mientras los elementos *comprobar* y *correcto* hacen referencia al estado en el que se encuentran, estos dos últimos de *ayuda* y *error* deben aportar pistas o guías para los estados posteriores (especificados en *post*).

Finalmente el estado se cierra con la etiqueta `</estado>` y se pasa al siguiente si lo hubiera.

Hay casos excepcionales que permiten el acceso a estados de forma global, es decir, crear un estado que pueda ser accedido desde cualquier otro y que para ello no necesite estar incluido en *post*. Es más, el estado no variará, sólo se mostrará por pantalla el contenido de *correcto* del estado global y se mantendrá el estado anterior. Para crear un estado global basta con poner en la línea:

```
<estado id="1" codigo="DNA">
```

el *id* como una letra en lugar de un número. Por ejemplo:

```
<estado id="A" codigo="DNA">
```

Con esto haríamos al estado A accesible desde cualquier otro, siempre y cuando se introdujese el código adecuado.

Por último queda explicar el comportamiento de las funciones externas. En el simulador es posible añadir funciones creadas como complemento a las acciones normales de los ejercicios. Estas funciones se introducen en la aplicación siguiendo las instrucciones del capítulo 3 de este documento (lo explicaremos detenidamente en el siguiente apartado, *Cómo crear ejercicios*), y una vez hecho esto pueden ser llamadas desde los estados de los ejercicios.

Para añadir una función se puede hacer en dos elementos: *comprobar* y *correcto*. Ninguno de los dos es obligatorio, pero la necesidad de incluirlos dependerá en cada

caso del funcionamiento de la función en sí.

En la comprobación se haría de la siguiente manera:

```
<comprobar>
  <codigo>DAC</codigo>
  <args>
    <a option="eq" function="DAC"/>
  </args>
</comprobar>
```

La etiqueta *codigo* es opcional, pero la forma de pedir la función es la misma siempre. Se mantiene la etiqueta *option* y se añade la etiqueta *function* pidiendo el nombre de la función. En este ejemplo la etiqueta *<a>* se abre y se cierra sin tener contenido, añadiendo la barra */* al final. Es equivalente a:

```
<a option="eq" function="DAC"></a>
```

Ahora que ya sabemos trabajar con funciones sólo queda conocer un atributo más:

```
<estado id="4" codigo="DNE" final="true">
```

Ésta es una línea de inicio de estado, con el atributo *final="true"*, que indica que éste es el último estado del ejercicio, y por lo tanto si se hace logout en este momento la función encargada de poner las notas sabrá que se ha llegado al final correctamente. Sólo se debe añadir *final="true"* si se quiere hacer calificable el ejercicio.

## 5. Cómo crear ejercicios

Los ejercicios se encuentran en la carpeta EjerciciosXML del directorio de la aplicación dentro de Tomcat. Para crear un ejercicio nuevo basta con crear un nuevo archivo *“.xml”* y añadir la línea:

```
<ejercicio id="0" nombre="Simulador tema 2">
```

Dentro de esta etiqueta vendrán los diferentes estados siguiendo la forma ya explicada en el apartado anterior. El *id* del ejercicio no es necesario, mientras que el *nombre* indica el nombre del ejercicio (*actividad Offline* en este caso) dentro de Moodle al cual se le debe poner la nota. Este atributo junto con *final*, visto anteriormente, son los que indican qué nota recibirá el alumno. Si se desea que el ejercicio no sea calificable se debe omitir este atributo.

Tras la etiqueta de comienzo del ejercicio viene un elemento especial, que es *inicio*. Aquí se debe incluir el texto que se quiera mostrar nada más comenzar el ejercicio, previo al login del usuario.

<inicio>

Texto de inicio, enunciado del ejercicio.

</inicio>

Los ejercicios requieren dos estados “no globales” como mínimo para funcionar. El primero por convenio se es *Jl* para partir del estado 0:

```
<estado id="0" codigo="Jl">

    <post>
        <p>1</p>
        <p>2</p>
    </post>

    <comprobar>
        <codigo>Jl</codigo>
    </comprobar>

    <error>
        Texto de error para ir a 1 o a 2
    </error>

    <ayuda>
        Ayuda para ir a 1 o a 2
    </ayuda>

</estado>
```

Este es el estado 0 base para todos los ejercicios. El siguiente estado obligatorio puede tener cualquier contenido. Adicionalmente pueden existir tantos estados globales como se quiera, pero al menos debe de haber dos estados normales, siendo uno obligatoriamente el “0”.

Al finalizar todos los estados el ejercicio se debe cerrar con la etiqueta de cierre *</ejercicio>*.

Por último, se debe añadir un enlace al ejercicio en la página desde la que se quiera acceder, indicada en el enlace del ejercicio de Moodle. Si se desea se puede crear una nueva página. Esta página debe contener el script:

```
<script language="JavaScript">
    function openShell(fileXML) {
        var left = (screen.width - 600) / 2;
        var top = (screen.height - 400) / 2;
        var w = window.open("console.jsp?archivo=" + fileXML,
            "wincmd_shell", "width=700,height=450,left=" + left + ",top=" + top +
            ",scrollbars=yes,status=no,resizable=no");
    }
</script>
```

Este script permite la apertura de una página guardando el estilo de terminal. Si la página se encuentra en un lugar que no sea la carpeta raíz de la aplicación en Tomcat se debe añadir la dirección delante de `console.jsp`, por ejemplo: `../console.jsp` si el archivo se encuentra en una subcarpeta dentro de la principal, o <http://localhost:8080/SimAmadeus/console.jsp> si queremos evitar problemas accediendo así al servidor en local. Después se añade el ejercicio de la siguiente manera:

```
<a href="javascript:openShell('archivo');"><b>nombre</b></a>
```

Donde *archivo* es el nombre del archivo del ejercicio sin la terminación *xml*, y *nombre* es el nombre que queremos que se muestre en la página web como texto del enlace.

## 6. Cómo ver la calificación de los alumnos

Si todo ha funcionado correctamente los alumnos realizarán los ejercicios de Moodle y del simulador y sus notas se irán añadiendo automáticamente a su perfil. Para ver las calificaciones y comprobar que todo está funcionando correctamente se puede acceder desde la página del curso, en el panel de Administración, normalmente situado a la izquierda, bajo el nombre Calificaciones.

Es una buena forma de comprobar si los parámetros de conexión y la configuración del simulador están funcionando correctamente.

## MANUAL TÉCNICO/INSTALACIÓN

### 1. Requisitos

Ya hablamos de los requisitos del sistema al comienzo del capítulo 3. Las limitaciones vienen impuestas principalmente por la cantidad de usuarios y el tráfico esperado en el servidor. Apache, MySQL y Tomcat funcionarán adecuadamente en cualquier ordenador que no tenga excesiva antigüedad. Para información más específica es recomendable visitar las páginas web de las aplicaciones necesarias.

### 2. Enlaces para obtener el software

A continuación listamos los enlaces a las páginas oficiales de las aplicaciones requeridas para la instalación del curso:

Apache + PHP: <http://www.apache.org/>

MySQL: <http://www.mysql.com>

Tomcat: <http://tomcat.apache.org/>

Moodle: <http://moodle.org>

### 3. Propuesta AMP

Como también explicamos en el capítulo 3 se puede optar por una instalación rápida usando soluciones AMP<sup>40</sup>, que integran en una misma instalación Apache + PHP + MySQL preconfigurados, además de poder incluir algún paquete más (python, perl, tomcat, etc.). Es una solución tremendamente útil porque reduce el tiempo dedicado a la instalación, y configuración del servidor. Es posible que dependiendo de las necesidades se requiera una configuración más específica, pero la que viene por defecto suele ser suficiente para la mayoría de entornos.

La más extendida es XAMPP<sup>41</sup>, pero hay muchas otras que pueden ofrecer prestaciones diferentes y más acordes con las necesidades del servidor. Esta en particular no incluye Tomcat, que es necesario para los ejercicios del simulador y debe ser instalado posteriormente.

### Instalación MySQL

En la dirección web de MySQL se encuentran diferentes opciones de descarga e instalación para diversos Sistemas Operativos. Si se usan distribuciones de Linux que usen descargas basadas en repositorios (como Debian, Ubuntu, ArchLinux, etc.) se puede instalar desde ahí, pero primero es necesario comprobar la versión disponible, puesto que en el caso de que se busque la más reciente es posible que sólo se encuentre en la web oficial.

Es aconsejable instalar MySQL en un disco duro donde se disponga de al menos 4 GB de espacio libre, puesto que es una aplicación que hace un uso intensivo del espacio y, a pesar de que Moodle no tiene una carga excesiva, si el servidor recibe una gran cantidad de usuarios puede alcanzar un tamaño importante.

---

<sup>40</sup> Acrónimo de Apache+MySQL+PHP. Habitualmente se usa acompañado de una L – LAMP para indicar la instalación en entornos Linux.

<sup>41</sup> Disponible en: <http://www.apachefriends.org/es/xampp.html> [Consulta 4-05-09]

Durante la instalación se pedirá un puerto de acceso, habitualmente se deja el que viene dado por defecto que es el 3306, pero se puede cambiar si hubiera algún conflicto. Además se requiere un password para el usuario root (es recomendable no dejarlo en blanco, pues sería un potencial agujero de seguridad).

Posteriormente habrá que seguir las instrucciones de Moodle para crear un usuario con acceso específico a la base de datos de la plataforma de e-learning.

## **5. Instalación Apache**

Al igual que dijimos sobre MySQL, en la web de Apache se encuentran distintas opciones de descarga e instalación, siendo posible también su instalación desde repositorios.

La instalación en Windows no requiere de atenciones especiales, por contra en Linux, con las restricciones por permisos de usuario para el acceso a carpetas, puede ser necesario crear una carpeta específica para Moodle llamada “moodledata” en el directorio inmediatamente superior al de las páginas servidas por Apache<sup>42</sup>. Para otros Sistemas Operativos seguir las instrucciones específicas de cada uno.

## **6. Instalación PHP**

En las nuevas versiones de Apache viene incluido PHP y configurado para funcionar desde la instalación. En caso de que surgiera cualquier problema, en la página web de PHP se encuentran respuestas y soluciones a las incidencias más habituales.

## **7. Instalación de Tomcat**

Es prácticamente igual a la explicada para Apache. Puede que requiera especificar un nombre y contraseña para la gestión de Tomcat (varía en función de la versión, puede incluir una contraseña vacía por defecto). En cualquier caso no es estrictamente necesario.

Se aconseja mantener Tomcat y Apache en la misma estructura de carpetas para tener un mejor acceso para añadir o modificar elementos.

---

<sup>42</sup> Si el directorio usado por Apache es /var/www/ la carpeta se encontraría en /www/moodledata. En cualquier caso se recomienda comprobar primero los permisos de las carpetas y probar la instalación.



## 8. Instalación de Moodle

Siguiendo las instrucciones ofrecidas en su página web no debería haber ningún problema, el proceso es automático e independiente del SO (se instala a través de Apache y MySQL). Es necesario prestar atención a lo que comentamos sobre la carpeta “moodledata” en el apartado sobre la instalación de Apache.

Aquí también se pedirán una serie de datos, que debemos conservar para la posterior instalación del curso y del simulador, que son el usuario y contraseña para acceder a la base de datos de Moodle, y el prefijo que llevarán las tablas de dicha base de datos (por defecto es “mdl\_”). Estos datos se usarán luego para conectarse a la base de datos para comprobar los usuarios y añadir las notas.

Con esto hecho Moodle estará funcionando y podremos pasar a instalar el curso.

## 9. Instalación del Curso en Moodle

En Moodle, dentro del panel de administración accesible como editor del sitio, se encuentra la opción “Restaurar”. Desde aquí se puede subir un archivo, en este caso el archivo con la copia del curso de *Introducción a Amadeus*, y seleccionar la acción de *Restaurar*. Entre las varias opciones que ofrece el sistema y que se podrán escoger según las preferencias, se encuentra una que señala si se quiere instalar sobre un curso o crear uno nuevo. Hay que elegir ésta con cuidado para no “pisar” un curso anterior o repetir uno ya creado.

Una vez finalizado el proceso el curso estará instalado y listo para recibir estudiantes.

Aquí hay otros datos que nos interesa apuntar: el nombre del curso, el nombre corto y los nombres que reciben las actividades Offline incluidas en los temas (vienen establecidas como Ejercicios de Simulador del tema X, pero si se cambia el nombre es necesario apuntarlo junto con los demás datos que hemos ido especificando).

## 10. Instalación de la aplicación en Tomcat

Para instalar la aplicación en Tomcat simplemente hay que copiar el archivo “.war” o la carpeta (el nombre actual es SimAmadeus, pero puede cambiarse a otro si se prefiere sin afectar al funcionamiento del simulador) dentro del directorio “webapps” de Tomcat. Si se copia el “.war” el propio servicio se encargará de desplegarlo y colocarlo en su carpeta correspondiente. Si se prefiere copiar la carpeta estará

inmediatamente accesible.

Para acceder después a la aplicación se usa el puerto 8080 del servidor de la forma <http://servidor:8080/SimAmadeus> si fuera SimAmadeus el nombre usado. En cualquier caso el acceso a los ejercicios se realiza mediante enlaces creados para interactuar con Moodle, por lo que sólo se recomienda acceder usando este método para realizar pruebas de funcionamiento. En este caso es conveniente recordar que las funciones de conexión con Moodle disponen de un usuario por defecto llamado “usuariodeprueba” con el password “passworddeprueba”, mediante el que se puede acceder para probar el funcionamiento de los ejercicios.

## 11. Añadir funciones al simulador

En el manual del profesor ya hemos visto cómo usar funciones del simulador para mostrar en determinados pasos del ejercicio, ahora vamos a ver cómo añadir dichas funciones al simulador.

La función creada en java de implementar la interfaz *CargaFunciones* y estar incluida en el paquete de la aplicación dentro del entorno de Tomcat. Una vez pueda ser referenciada se debe cargar en la clase Funciones. La forma en la que se realiza esto actualmente es desde la página *responder.jsp*.

Bajo el comentario:

```
//Carga de funciones
```

Se encuentran las funciones cargadas actualmente. La forma de hacerlo es poniendo:

```
sim_entrada.cargarFuncion("nombre",new nombreFuncion());
```

Donde el *nombre* entre comillas será el nombre por el que podrá ser llamada dentro de los ejercicios, y *nombreFuncion()* será la llamada al constructor, para lo que requiere el nombre de la función tal y como se haya creado.

Hecho esto ya se puede utilizar la función, siempre que cumpla con los pasos establecidos.

### **Datos de configuración de la BD (.ini) para la función de Sign Out**

Los datos que hemos ido recogiendo sirven para indicar a las funciones de login y logout dónde buscar los datos del usuario e introducir las notas de los ejercicios de

simulación. Las funciones buscarán los datos de conexión en el archivo *mysql\_connect.ini*, y las líneas que se deben cambiar son:

*Para indicar la IP o la dirección web en la que se encuentra MySQL, localhost si se encuentra en el mismo servidor que Tomcat.*

MySQLserver\_ip :: localhost

*El nombre de la BD dentro de MySQL que contiene los datos de Moodle. Normalmente se llama "moodle", como indica el ejemplo.*

BD\_name :: moodle

*El usuario y contraseña para acceder a dicha base de datos. En este ejemplo se usa "root" de usuario y "pswr" de contraseña, pero es preferible no usar el usuario "root" si no el específico de Moodle, o incluso uno creado específicamente para el caso.*

MySQL\_user :: root

MySQL\_PSW :: pswrd

*El prefijo de las tablas de la BD. El símbolo # indica que la línea está comentada, por lo que no tiene una función específica más allá de informar de cual es el prefijo que debe usarse en las tablas que se especifican en el mismo archivo.*

#moodle\_prefix :: mdl\_

*Finalmente el nombre largo y corto del curso para identificarlo frente a los otros cursos que estén alojados en la plataforma.*

course\_name :: Curso de AMADEUS

course\_shortcode :: AM

*Tras estas líneas siguen las que indican las tablas que deben ser accedidas. Éstas deben llevar un prefijo según está indicado en el comentario de más arriba.*

Todas estas variables son indispensables para el correcto funcionamiento del simulador, siempre que se usen las funciones para el login y logout, y se mantenga

para realizar las conexiones MySQL la clase Conector creada para ello, que es la que usa en última instancia los datos de *mysql\_connect.ini*.

## BIBLIOGRAFÍA

Manuales, monografías, artículos de publicaciones periódicas.

AMARTINO, Mariano. "AJAX un nuevo acercamiento a Aplicaciones Web". Disponible en <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-web.php> [Consulta 4-05-09]

BONEU, Josep M. "Plataformas abiertas de e-learning para el soporte de contenidos educativos abiertos". *Revista de Universidad y Sociedad del Conocimiento*, 2007, vol. 4, nº 1. Disponible en <http://www.uoc.edu/rusc/4/1/dt/esp/boneu.pdf> [Consulta 03-05-09]

COLE, Jason R. *Using Moodle: teaching with the popular open source course management system*. Sebastopol, CA: O'Reilly, 2005. 219 p.

GARCÍA PEÑALVO, Francisco José. "Estado actual de los sistemas e-learning". *Teoría de la Educación: educación y cultura en la sociedad de la información*, .2005, vol. 6, nº 2. Disponible en: [http://campus.usal.es/~teoriaeducacion/rev\\_numero\\_06\\_2/n6\\_02\\_art\\_garcia\\_penalvo.htm](http://campus.usal.es/~teoriaeducacion/rev_numero_06_2/n6_02_art_garcia_penalvo.htm) [Consulta 3-05-09]

EGEA GARCÍA, Carlos; Alicia Sarabia Sánchez, Alan Chutter. *Diseño de páginas web accesibles*. Biblioteca Digital de Jesús David. (E-book's, Manuales de Desarrollo Web) Disponible en: <http://www.jesusda.com/docs/ebooks/index.html#manuales-desarrollo-web>

Entrevista a Martin Dougiamas, *El País*, 4 de diciembre de 2008. Disponible en: [http://www.elpais.com/articulo/portada/Dougiamas/creador/Moodle/Internet/cambia/educacion/elpepiscib/20081204elpepiscibpor\\_3/Tes](http://www.elpais.com/articulo/portada/Dougiamas/creador/Moodle/Internet/cambia/educacion/elpepiscib/20081204elpepiscibpor_3/Tes) [Consulta 3-05-09]

FALKNER, Jayson. *Desarrollo Web con JSP: fundamentos*. Madrid: Anaya Multimedia, 2002. 846 p. (Anaya Multimedia/ Wrox)

FROUFE QUINTAS, Agustín. *Java 2: manual de usuario y tutorial*. Madrid: Ra-Ma, 2008. 674 p.

HANNA, Phil. *JSP: manual de referencia*. Madrid: McGraw-Hill Interamericana de

España, 2002. 776 p.

HAROLD, Elliotte Rusty. *XML bible*. 2ª ed. New York: Hunfry Minds, 2001. 1206 p.

MARCELO GUARDIA, Marcelo. Ventajas y desventajas del Moodle aplicado en el aprendizaje cooperativo. Entrada de blog: TIC en Educación Básica- Grupo 4, Dirección de Informática Académica de la Pontificia Universidad Católica del Perú. Disponible en: <http://blog.pucp.edu.pe/item/26621> [Consulta 5-05-09]

MELLADO, Javier. Diez razones para usar AJAX. *AJAX hispano*. 17 de enero de 2006. Disponible en <http://www.ajaxhispano.com/diez-razones-para-usar-AJAX.html> [Consulta 10-05-09]

MOLIST, Mercè. Moodle llena la geografía educativa española de campus virtuales. *El País*, 4 de diciembre de 2008. Disponible en: [http://www.elpais.com/articulo/portada/Moodle/llena/geografia/educativa/espanola/campus/virtuales/elpepusupcib/20081204elpepusupcibpor\\_1/Tes/](http://www.elpais.com/articulo/portada/Moodle/llena/geografia/educativa/espanola/campus/virtuales/elpepusupcib/20081204elpepusupcibpor_1/Tes/) [Consulta 5-05-09]

NICHOLS, Mark. "E-Learning in Context". *E-Primer Series*. Disponible en <http://akoaootearoa.ac.nz/sites/default/files/ng/group-661/n877-1---e-learning-in-context.pdf> [Consulta 3-05-09]

PEMBERTON, Steven [et al.]. XHTML™ 1.0. The Extensible HyperText Markup Language (2ª ed.): a Reformulation of HTML 4 in XML 1.0. *W3C recommendation*, 26 enero 2000. Disponible en: <http://www.w3.org/TR/xhtml1/> [Consulta 7-05-09]

RICE, William H. *Moodle: e-learning course development: a complete guide to successful learning using Moodle*. Birmingham, UK: Packt Publishing, 2006. 236 p.

ROSENBERG, Marc Jeffrey. *E-learning: strategies for delivering knowledge in the digital age*. New York: McGraw-Hill, 2001. 344 p.

SAGASETA, Fernando. "Amadeus sigue creciendo entre las agencias: las nuevas herramientas no acaban de hacer mella en la distribución tradicional". *Editur: El semanario profesional del turismo*, 2007, nº 2447, p. 17

SÁNCHEZ ALLENDE, Jesús. *Programación en Java*. New York: McGraw Hill, 2009. 257 p.

TRIGOS GARCÍA, Esteban. *JSP*. Madrid: Anaya Multimedia, 2001. 303 p. (Guías

prácticas)

VILA, Jordi. "E-learning: la revolución ha comenzado". Tomado de eLearning Workshops (Comunidad de eLearning). Disponible en: <http://www.elearningworkshops.com/modules.php?name=News&file=article&sid=323> [Consulta 03-05-09]

Páginas y portales web.

API de Java 5. Disponible en: <http://java.sun.com>

*Desing Elements and Principles*. Universidad de California, San Diego. Disponible en: <http://gort.ucsd.edu/preseduc/design.htm>

*Estandarización web*. DocumentoWeb. Disponible en: <http://www.documentoweb.com/HTML/estandarizacion-web.php>

*Precise Java*. Disponible en: <http://www.precisejava.com/>

SEIDLER, Kai. WAMPP. ApacheFirends <http://www.apachefriends.org/en/xampp.html>

Using MySQL with Java. MySQL Java. Disponible en: <http://dev.mysql.com/usingmysql/java/>

w3schools. Disponible en: <http://www.w3schools.com>

*What is Web 3.0? Semantic Web and other Web 3.0 concepts explained in plain English*. Disponible en: <http://www.labnol.org/internet/web-3-concepts-explained/8908/>

Página oficial de Apache. Disponible en: <http://www.apache.org/>

Página oficial de PHP. Disponible en: <http://www.php.net>

Página oficial de MySQL. Disponible en: <http://www.mysql.com>

Página oficial de Tomcat. Disponible en: <http://tomcat.apache.org/>

Página oficial de Moodle. Disponible en: <http://moodle.org>





